MATEMATICI DISCRETE APLICATE

Probleme

Compilație realizată de
Gheorghe M.Panaitescu

Catedra Automatică si calculatoare
Universitatea "Petrol-Gaze" Ploiesti
2007

I N T R O D U C E R E

Prezenta compilatie este o traducere cu adaptări minore a problemelor date ca teme de casă studentilor care au urmat cursul de Matematici discrete tinut în primăvara anului 2005 la University of California, Berkeley, de **Michael J.Clancy** si **David Wagner**.

Am adăugat câteva probleme despre grafuri fromulate pe baza cursului predat de **Marc Pomplun** la University of Massachusetts at Boston.

În multe puncte, pentru întelegerea unor detalii si transpunerea lor corectă si inteligibilă în limba română am apelat la cursul de Matematici discrete predat de **David A.Santos** la Community College of Philadelphia.

Toate aceste surse sunt accesibile public pe site-urile universitătilor mentionate.

Aceste Probleme si aplicatii, de regulă gata rezolvate acompaniază lectiile de Matematici discrete grupate într-un volum separat si preluate din aceleasi surse.

Problemele nu sunt grupate pe capitole/lectii precum cele din volumul celălalt, cel care prezintă aspecetele teoretice. Se consideră că cititorul nu va avea dificultăti în a asocia subiectele teoretice cu aplicatiile prezentate în acest volum.

1. **Reguli de inferență**
   Pentru fiecare din punctele următoare, definiți simbolurile pentru fiecare propoziție simplă. Apoi scrieți forma logică a expresiei. Dacă forma expresiei corespunde unei reguli de inferență cunoscute, spuneți care este aceea. Dacă nu, arătați că demonstrația este corectă uzând de tabele de adevăr.
   (a) Voi lucra întâi această temă si voi avea satisfacție lucrând-o. Asadar, voi lucra întâi această temă.
   P = "voi lucra întâi la această temă"
   Q = "voi avea satisfacție lucrând-o"

   Soluție: $\dfrac{P \wedge Q}{P}$ eliminare-de-si

   (b) Azi este mai cald de 33 grade sau poluarea este periculoasă. Azi este mai puțin de 33 grade asadar poluarea este periculoasă.
   P = "este mai cald de 33 de grade"
   Q = "poluarea este periculoasă"

   Soluție: $\dfrac{P \vee Q, \neg P}{Q}$ eliminare-de-sau

   (c) Ioana va proceda anul viitor la începerea scolii. Asadar, Ioana va proceda anul viitor la începerea scolii sau ea va fi somer.
   P = "Ioana va proceda anul viitor la începerea scolii"
   Q = "ea va fi somer"

   Soluție: $\dfrac{P}{P \vee Q}$ introducere-de-sau

   (d) Dacă lucrez toată noaptea la temă, voi răspunde la toate exercițiile. Dacă răspund la toate exercițiile, voi înțelege materialul cursului. Asadar, dacă lucrez toată noaptea la această temă, voi înțelege materialul cursului.
   P = "lucrez toată noaptea la temă"
   Q = "voi răspunde la toate exercițiile"
   R = "voi înțelege materialul cursului"

   Soluție: $\dfrac{P \Rightarrow Q, Q \Rightarrow R}{P \Rightarrow R}$ silogism de ipoteze

2. Reamintim că $\mathbf{N} = \{0, 1, \ldots\}$ este mulțimea numerelor naturale si $\mathbf{Z} = \{\ldots, -1, 0, 1, \ldots\}$ este mulțimea numerelor întregi.
   (a) Se defineste $P(n) = \forall\, m \in \mathbf{N},\ m < n \Rightarrow \neg(\exists\, k \in \mathbf{N},\ n = mk \wedge k < n)$. Concis, pentru care numere $n \in \mathbf{N}$ $P(n)$ este adevărată?

Soluție: Este adevărată pentru numerele prime, pentru 0 si pentru 1.

(b) Rescrieți următoarea propoziție într-un mod care elimină toate negațiile ("¬", "≠") propoziția rămânând echivalentă.

$\forall\ i.\ \neg\forall\ j.\ \neg\exists\ k.\ (\neg\exists\ l.\ f(i,j) \neq g(i,l))$

Soluție: $\forall\ i.\ \exists\ j.\ \forall\ k.\ (\forall\ l.\ f(i,j) = g(i,l))$

(c) Dovediți sau dovediți contrarul: $\forall\ m \in \mathbf{Z}.\ \exists\ n \in \mathbf{Z}.\ m \geq n$.

Soluție: Adevărat

Fie $P(m) = \exists\ n \in \mathbf{Z}.\ m \geq n$. Se ia un $m \in \mathbf{Z}$, arbitrar. $n = m$ satisface $m \geq n$, asadar $P(m)$ este adevărată. Deoarece $P(m)$ este adevărată pentru un $m \in \mathbf{Z}$, arbitrar, ea este adevărată pentru orice $m \in \mathbf{Z}$.

(d) Dovediți sau dovediți contrarul: $\exists\ m \in \mathbf{Z}.\ \forall\ n \in \mathbf{Z}.\ m \geq n$.

Soluție: Fals

Fie $P(m) = \forall\ n \in \mathbf{Z}.\ m \geq n$. Se ia un $m \in \mathbf{Z}$, arbitrar. Se ia $n = m + 1$. Prin definiția întregilor, n este si el în $\mathbf{Z}$. Dar $m < n$. Asadar $P(m)$ este falsă pentru un $m \in \mathbf{Z}$, arbitrar. Deoarece $P(m)$ este falsă pentru un $m \in \mathbf{Z}$, arbitrar, ea este falsă pentru orice $m \in \mathbf{Z}$.

3. Alina si Mihai joacă sah. Alina este la prima mutare. Dacă $x_1, \ldots, x_n$ este o secvență de mutări posibile (adică Alina face mutarea x_1, Mihai face mutarea x_2 s.a.m.d.), punem $W(x_1, \ldots, x_n)$ propoziția care spune că după efectuarea acestei secvențe de mutări, Mihai este mat.

(a) Folosind cuantificatorii, scrieți propoziția conform căreia Alina face mat la a doua mutare, indiferent ce joacă Mihai.

Soluție: $\exists\ x_1.\forall\ x_2.\exists\ x_3.W(x_1, x_2, x_3)$

(b) La prima mutare, Alina are multe posibilități de a alege prima sa mutare si doreste a găsi una care îi permite a face mat adversarul la mutarea a doua. Formulați utilizând cuantificatorii, propoziția conform căreia x_1 nu este o astfel de mutare.

Soluție: $\exists\ x_2.\forall\ x_3.\neg W(x_1, x_2, x_3)$ sau $\neg\forall\ x_2.\ \exists\ x_3.W(x_1, x_2, x_3)$

4. Ioana este fie un om de cuvânt fie dimpotrivă, o persoană fără cuvânt. Un om de cuvânt spune totdeauna adevărul si numai adevărul: omul fără cuvânt spune totdeauna lucruri false si numai lucruri false. Cineva o întreabă pe Ioana: "Esti om de cuvânt?". Ea răspunde: "Dacă sunt om de cuvânt îmi voi mânca pălăria".

(a) Trebuie Ioana să-si mănânce pălăria?

Soluție: Da

(b) Să punem această problemă într-o logică propozițională. Introduceți propozițiile următoare:

P = "Ioana este om de cuvânt"

Q = "Ioana îsi va mânca pălăria"

Traduceți ceea ce se spune în enunț în logica propozițiilor.

Soluție: $P \Rightarrow (P \Rightarrow Q)$

$\neg P \Rightarrow \neg(P \Rightarrow Q)$

(c) Utilizând demonstrația prin enumerare, dovediți că răspunsul din prima parte decurge din premisele scrise în partea a doua (fără reguli de inferență).

Soluție: Să scriem tabelul de adevăr pentru implicațiile de mai sus.

P	Q	$P \Rightarrow Q$	$P \Rightarrow (P \Rightarrow Q)$	$\neg P \Rightarrow \neg(P \Rightarrow Q)$
F	F	T	T	F
F	T	T	T	F
T	F	F	F	T
T	T	T	T	T

Căutând în tabelul de adevăr când sunt cele două implicații ambele adevărate, găsim că ele cer ca P si Q să fie ambele adevărate. Asadar, Ioana trebuie să-si mănânce pălăria.

5. Pentru fiecare din afirmațiile de mai jos stabiliți adevărul sau falsitatea.

(a) Orice întreg pozitiv poate fi exprimat ca suma a două pătrate perfecte (un pătrat perfect este pătratul unui întreg; 0 poate fi utilizat în sumă).

Soluție: Fie $P(n) = \exists\, a \in \mathbf{Z}.\ \exists\, b \in \mathbf{Z}.\ n = a^2 + b^2$. Afirmația de demonstrat este $\forall\ n \in \mathbf{Z}.\ P(n)$.

Fals.

Demonstrație prin contraexemplu.

Fie $n = 3$. Deoarece pătratul unui întreg este totdeauna un întreg nenegativ, a^2 si b^2 trebuie să fie nenegative. Deoarece suma a două numere nenegative este mai mare sau egală cu fiecare termen, $a^2 \leq n$ si $b^2 \leq n$. Deoarece $n = 3$, a^2 si b^2 trebuie să fie 0 sau 1. Dar nici 0 + 0, nici 0 + 1, nici 1 + 1 nu dau 3. Asadar $P(3)$ este falsă.

(b) Pentru orice numere raționale a si b, a^b este tot rațional.

Soluție: Fie $\mathbf{Q}$ mulțimea tuturor numerelor raționale. Fie $P(a, b) = a^b \in \mathbf{Q}$. Afirmația al cărui adevăr trebuie dovedit este $\forall\ a \in \mathbf{Q}.\forall\ b \in \mathbf{Q}.\ P(a, b)$.

Fals.

Demonstrație prin contraexemplu.

Fie $a = 2$, $b = 1/2$. $a^b = 2^{1/2} = \sqrt{2}$ prin definiția rădăcinii pătrate. Prin demonstrația din textul cursului, $\sqrt{2} \notin \mathbf{Q}$. Astfel $P(2, 1/2)$ este falsă.

1. Practica demonstratiilor de propozitii

Demonstrati propozitia

$$(P \Rightarrow (Q \Rightarrow (P \wedge Q))) \Rightarrow (\neg(P \wedge Q) \Rightarrow (P \Rightarrow \neg Q))$$

Demonstratia poate include demonstratii subsidiare sau leme (similar functiilor "helper" din Scheme). Se pot totodată utiliza două tipuri de *presupuneri*. Dacă încercati să arătati o implicatie $E_1 \Rightarrow E_2$, o strategie validă de demonstrare este de a presupune E_1 si apoi a utiliza această presupunere pentru a demonstra E_2 (dar e permis numai a presupune E_1 în scopul demonstrării implicatiei; asumarea lui E_1 nu este permisă pentru a "supravietui" pe parcursul demonstratiei că $E_1 \Rightarrow E_2$). De asemenea e permisă asumarea lui $\neg E$ pentru a demonstra o expresie E (dar din nou această presupunere nu poate rămâne pe parcursul corpului demonstratiei lui E).

Demonstratia poate utiliza numai următoarele reguli de inferentă:

- Modus ponens: din expresiile E_1 si $(E_1 \Rightarrow E_2)$ se poate infera E_2.
- Modus tollens: din expresia $\neg E_2$ si $(E_1 \Rightarrow E_2)$ se poate infera $\neg E_1$.
- Constructia cu implicatie: din expresia E_2 se poate infera $(E_1 \Rightarrow E_2)$ pentru orice expresie E_1.
- Contradictia: din expresiile E si $\neg E$ se poate infera orice expresie.
- Eliminarea-de-si: din expresia $E_1 \wedge E_2$ se poate infera E_1 (sau E_2).
- Echivalenta logică: din expresia E_1 se poate infera expresia E_2 dacă $E_1 \equiv E_2$ (dacă E_1 si E_2 sunt echivalente logic, cum s-a definit în curs).

Solutie:

De demonstrat: $(P \Rightarrow (Q \Rightarrow (P \wedge Q))) \Rightarrow (\neg(P \wedge Q) \Rightarrow (P \Rightarrow \neg Q))$

Demonstratie:

1. Se admite că $P \Rightarrow (Q \Rightarrow (P \wedge Q))$, lucru permis pentru demonstrarea teoremei.
2. Lemă: $\neg(P \wedge Q) \Rightarrow (P \Rightarrow \neg Q)$

   Demonstratie:

   2.1. Se admite $\neg(P \wedge Q)$, pentru a demonstra lema a doua

   2.2. Lemă: $P \Rightarrow \neg Q$

   Demonstratie:

   2.2.1. Se admite P

   2.2.2. Lema: $\neg Q$

   Demonstratie:

   2.2.2.1.Se presupune $\neg\neg Q$

   2.2.2.2.Q (prin aplicarea $\neg\neg Q \equiv Q$ afirmatiei 2.2.2.1.)

   2.2.2.3.$Q \Rightarrow (P \wedge Q)$ (modus ponens aplicat pasilor 2.1.1. si 1.)

   2.2.2.4.$P \wedge Q$ (modus ponens aplicat pasilor 2.2.2.3. si 2.2.2.4.)

2.2.2.5.$\neg Q$ (contradictie rezultată din pasii 2.1. si 2.2.2.4.)

□

2.2.3. $P \Rightarrow \neg Q$ (constructie implicativă aplicată pasului 2.2.2.)

□

2.3. $\neg(P \wedge Q) \Rightarrow (P \Rightarrow \neg Q)$ (constructie implicativă aplicată pasului 2.2.)

□

3. $(P \Rightarrow (Q \Rightarrow (P \wedge Q))) \Rightarrow (\neg(P \wedge Q) \Rightarrow (P \Rightarrow \neg Q))$ (constructie implcativă aplicată pasului 2.).

□

2. **Inductia simplă**.

Demonstrati că $2n < n!$ pentru orice întreg $n \geq 4$.

Solutie:

- Cazul de bază: $P(4)$ este propozitia $2^4<4!$ si este adevărată deoarece $16 < 24$.
- Pasul inductiv: demonstrarea faptului că $P(k) \Rightarrow P(k + 1)$ pentru orice $k \geq 4$.

(a) Se presupune $P(k)$ adevărată adică $2^k < k!$.

(b) De demonstrat: $P(k + 1)$, adică $2^{k+1} < (k + 1)!$.

$$2^{k+1} = 2.2^k < 2.k! < (k + 1).k! = (k + 1)!$$

□

S-au utilizat succesiv definitia ridicării la putere, ipoteza inductivă, relatia $(k + 1) > 2$ si definitia functiei factorial.

3. Un alt mod de a multiplica

Se consideră următorul cod Scheme.

```
(define (product a1 a2)
      (product-helper a1 a2 0) )
(define (product-helper n1 n2 so-far)
      (cond
         ((= n1 1) (+ so-far n2))
          ((odd? n1) (product-helper (- n1 1) n2 (+ so-far
n2)))
         (else (product-helper (/ n1 2) (* 2 n2) so-far)) ) )
```

Rezultatul returnat de `product`, cu două argumente numere naturale, este produsul celor două numere. Deoarece ea implică numai adunări, scăderi, dublări si înjumătătiri, algoritmul furnizează o procedură practică de mutiplicare manuală a doi întregi (procedură utilizată cu cca. 2000 de ani î.C. de vechii egipteni; este cunoscută si ca "multiplicarea mujicului" deoarece vizitatorii Rusiei secolului al XIX-lea au constatat acolo largul uz al acestei metode).

Acelasi algoritm în pseudocod ar fi următorul:

Algorithm product(a_1, a_2):

1. return product-helper(a_1, a_2, 0).

Algorithm product-helper(n_1, n_2, s):

1. if $n_1 = 1$ then return $n_2 + s$.
2. if n_1 is odd, then return product-helper($n_1 - 1$, n_2, $n_2 + s$).
3. otherwise, return product-helper($n_1/2$, $2n_2$, s).

a. Aflati o formulă care leagă valorile `n1`, `n2` si `so-far` la începutul fiecărui apel la `product-helper`, de produsul numerelor `a1` si `a2`, argumentele initiale ale procedurii product.

Solutie: `a1 * a2 = n1 * n2 + so-far`

b. Utilizând relatia invariantă găsită, demonstrati că `product` returnează produsul argumentelor sale întregi si pozitive.

Solutie: Pentru a arăta că algoritmul lucrează, mai întâi se arată că invariantul `a1 * a2 = n1 * n2 + so-far` se mentine la începutul fiecărui apel la `product-helper`. Pentru a asimplifica notatia punem s pentru `so-far`.

Demonstratie: Fie $P(k)$ propozitia după care fie acest invariant se mentine la începutul fiecărui apel la `product-helper`, fie sunt mai putin de k apeluri la `product-helper`. Se foloseste inductia simplă după k.

În cazul de bază ($k = 1$), $n_1n_2 = 0 + a_1a_2$ chiar la primul apel la `product-helper` deoarece la acel apel $n_1 = a_1$, $n_2 = a_2$ si $s = 0$.

Pentru pasul inductiv, se admite că invariantul $n_1n_2 + s = a_1a_2$ se mentine la apelul k la `product-helper`. Fie n_1', n_2', s' argumentele pentru apelul $k + 1$ la acelasi `product-helper` (admitând că există unul; altminteri nu este nimic de dovedit). Vom demonstra că invariantul se mentine la apelul numărul $(k + 1)$, cu alte cuvinte, $n_1'n_2' + s' = a_1a_2$. Sunt de analizat trei cazuri, depinzând de valoarea lui n_1:

i. Dacă $n_1 = 1$, atunci nu există apelul $(k + 1)$ si avem finalul.

ii. Dacă n_1 este impar, atunci $n_1' = n - 1$, $n_2' = n_2$ si $s' = n_2 + s$, asa încât
$$n_1'n_2' + s' = (n_1 - 1)n_2 + n_2 + s = n_1n_2 + s = a_1a_2$$
unde pasul ultim provine din ipoteza inductivă.

iii. Dacă n_1 este par, atunci $n_1' = n_1/2$, $n_2' = 2n_2$ si $s' = s$, astfel că
$$n_1'n_2' + s' = (n_1/2).(2n_2) + s = n_1n_2 + s = a_1a_2$$
ceea ce completează demonstratia

□

Acum, vom argumenta faptul că `product-helper` se încheie totdeauna si cât timp $n_1 > 1$, n_1 descreste la un întreg totdeauna pozitiv în fiecare apel următor la `product-helper`.

Demonstratie: La primul apel, $n_1 = a_1 > 0$. Dacă $n_1 > 1$ si este impar, $0 < n_1 - 1 < n_1$ si $n_1 \in$ N. Dacă n_1 este par, $0 < n_1/2 < n_1$ si $n_1/2 \in$ N. Astfel, n_1 descreste si se mentine tot timpul întreg si pozitiv; prin buna ordonare a multimii numerelor naturale, în cele din urmă n_1 trebuie să atingă valoarea 1 si atunci `product-helper` se încheie.

□

Argumentatia arată că la ultimul apel la `product-helper` $n_1 = 1$. Deoarece invariantul $n_1n_2 + s = a_1a_2$ se mentine pentru toate apelurile, el se mentine si la ultimul apel si, în particular, la acest ultim apel devine $n_2 + s =$

a_1a_2 deoarece $n_1 = 1$. Asadar, ultimul apel la `product-helper` returnează produsul a_1a_2 corect.
În final se argumentează că `(product a1 a2)` returnează a_1a_2, care este răspunsul corect.
Demonstratie: Arătăm că prima invocare a rutinei `product-helper` returnează a_1a_2; apoi va urma afirmatia despre `(product a1 a2)`. Acestea sunt arătate prin inductie *inversă* după k. Cu alte cuvinte, fie $P(k)$ propozitia "apelul numărul k la `product-helper` (dacă există unul) returnează a_1a_2". Fie N indicele ultimului apel la `product-helper`. Folosim inductia simplă după $k = N, N - 1, \ldots, 1$. Cazul de bază $P(N)$ se verifică, cum s-a argumentat mai sus, ultimul apel la `product-helper` returnează a_1a_2. Pasul inductiv ($P(k + 1) \Rightarrow P(k)$) este facil deoarece cel de al k apel la `pruduct-helper` returnează ceea ce apelul numărul $(k + 1)$ la `product-helper` face.
(Alternativ, am putea utiliza inductia simplă pe j, cu predicatul $Q(j) = P(N + 1 - j)$. Totusi, această tratare ar putea fi o sursă de confuzie).
□
Am dovedit că `product` se termină totdeauna si returnează totdeauna răspunsul corect, astfel demonstratia de corectitudine a procedurii `product` este completă.
Comentariu: Cum ilustrează acest exemplu, invarianti sunt de mare ajutor în rationamentele asupra procedurilor recursive. Adesea este o problemă de stil bun în programare a documenta invariantii atunci când se scrie o procedură recursivă; aceasta ajută documentarea a ceea ce functia helper se presupune a executa, face mai usor accesul altor persoane la codul prim (altminteri ei ar trebui să refacă demonstratia prin inductie a corectitudinii) si încurajează autorul a verifica utilizarea corectă a inductiei în program. De pildă, algoritmul mujicului l-am fi putut scrie si implementa astfel:

```
;; Se presupune că a1 si a2 sunt întregi pozitivi
;; Returnează a1*a2
(define (product a1 a2)
  (product-helper a1 a2 0) )

;; Apelat numai din product si product-helper
;; Presupune n1*n2 + so-far = a1*a2
;; Returnează a1*a2
(define (product-helper n1 n2 so-far)
  (cond
    ((= n1 1) (+ so-far n2))
    ((odd? n1) (product-helper (- n1 1) n2 (+ so-far n2)))
    (else (product-helper (/ n1 2) (* 2 n2) so-far)) ) )
```

Această scriere face mai usoară verifiarea corectitudinii codului. Când este examinat un apel la o anumită functie trebuie că ne asigurăm că ea este apelată în (pre)conditii adecvate (presupunerile functiei apelate sunt

îndeplinite) si atunci documentarea (comentariile cu un termen mai comun) ne spune că suntem îndrituiti să concluzionăm asupra returului acelei functii. În particular, când vedem un apel la `f`, trebuie să privim la documentarea lui `f` si nu la implementarea lui `f`. La fel, la verificarea corectitudinii unei proceduri recursive trebuie observată numai respectarea invariantilor implicati.
La programarea în limbaje imperative de genul Java, este adesea necesar a scrie explicit invariantii asociati fiecărei bucle a programului.

4. Se consideră următorul cod Scheme. Un apel la `reverse` cu o listă de argumente returnează ca rezultat lista în ordine inversată,

```
(define (reverse L)
      (reverse-helper '( ) L) )
(define (revers-helper so-far L)
      (if (null? L) so-far
        (reverse-helper (cons (car L) so-far) (cdr L)) )
)
```

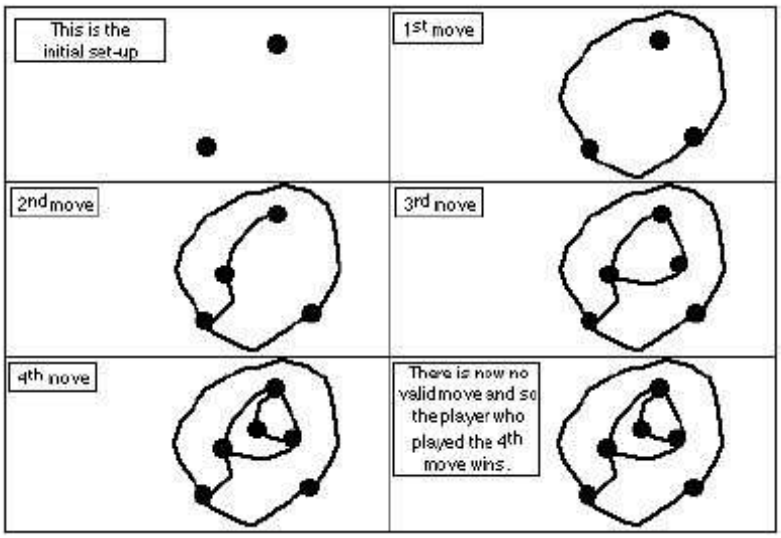
Demonstrati că `reverse` lucrează. Pentru asta e nevoie de câteva definitii. Dacă `L` = $(a_0a_1a_2 \ldots)$ atunci `(car L)` = a_0, `(cdr L)` = $(a_1a_2 \ldots)$ si `(cons x L)` = $(xa_0a_1a_2 \ldots)$.
Se poate dovedi că dacă `L` = $(x_0x_1x_2 \ldots x_N)$, atunci (`reverse L`) returnează $(x_Nx_{N-1} \ldots x_1x_0)$
Solutie: Se fixează o listă oarecare `L` = $(x_0x_1x_2 \ldots x_N)$. Vom examina ce se întâmplă când se execută `(reverse L)`. Fie $P(k)$ declaratia "argumentele celui de al k apel la `reverse-helper` sunt $(x_{k-2}x_{k-3} \ldots x_0)$ (ca valoare pentru `so-far`) si $(x_{k-1}x_k \ldots x_N)$ (ca valoare pentru `L`).". Demonstrăm că $P(k)$ este adevărată pentru orice $k \in \{2, 3, \ldots, N+1\}$.
Demonstratie: Se foloseste inductia după k (nu după N!). Cazul de bază ($k = 2$) are loc deoarece după primul apel de forma `(reverse-helper '( ) L)`, valoarea următoare pentru `so-far` este `(cons (car L) so-far)` = `(cons` x_0 `'())` = (x_0) si valoarea următoare pentru `L` este `(cdr L)` = $(x_1x_2 \ldots x_N)$.
Pentru pasul inductiv, se presupune $P(j)$ adevărată pentru un anumit întreg j. Atunci la apelarea $j + 1$ a rutinei `reverse-helper`, `so-far` este `(cons (car L) so-far)` = `(cons` $x_{j-1}(x_{j-2} \ldots x_0))$ = $(x_{j-1}x_{j-2} \ldots x_0)$ si `L` este `(cdr L)` = $(x_j \ldots x_N)$. Astfel, $P(j)$ implică pe $P(j + 1)$ (pentru $1 < j \leq N$).
□
Ca un corolar, pentru apelul $N + 1$ la `revers-helper`, argumentele sunt $(x_{N-1} \ldots x_0)$ pentru `so-far` si (x_N) pentru `L`. Următorul (si ultimul) apel la `reverse-helper` are forma

`(reverse-helper (`$x_N \ldots x_1x_0$`) '())`.

Deoarece `L` este vid (null) în acest apel, ultima chemare a rutinei `reverse-helper` returnează $(x_N \ldots x_1x_0)$.

În final, se poate demonstra că `(reverse L)` returnează $(x_N \ldots x_1 x_0)$, prin inductie inversată după *k* (numărul de apeluri la reverse-helper), întocmai ca la problema 3.
Comentariu: Din nou, la scrierea acestui cod, este o idee bună a documenta codul cu acesti invarianti. Pe măsură ce cititorul uzează de inductie, va găsi că aceste argumentatii inductive pot fi făcute mai usor mental, imediat ce aceti invarianti sunt observati. Este o cale de a verifica corectitudinea codului. În acest mod, obiceiul documentării este încurajat.

5. **Sprouts** (mlădite) este un joc pentru doi, care se joacă cu creionul pe hârtie. Se plasează pe hârtie un număr de puncte. Apoi jucătorii recurg pe rând la următoarele actiuni:



- trasarea unei linii între două puncte sau de la un punct la el însusi, linie care nu atinge/traversează vreo altă linie;
- punerea unui punct nou pe noua linie, punct care o separă în două linii.

Nici un punct nu poate avea mai mult de trei linii atasate. Câstigă jucătorul care poate face o ultimă actiune potrivit regulilor jocului.
Figura alăturată arată un astfel de joc.

a. Demonstrati că orice joc de Sprouts constă într-un număr finit de etape.

Solutie: În orice configuratie de puncte si linii, fiecare punct are cel mult trei linii atasate. Dacă un punct are atasate *d* linii, se defineste un *grad de neutilizare* a acelui punct, 3 – *d*. Totalul gradelor de neutilizare ale unei figuri este definit ca suma gradelor de neutilizare a puntelor din care ea este alcătuită. După fiecare miscare, gradul de neutilizare scade cu 1 pentru două puncte, ceea ce aduce o scădere a totalului cu 2 unităti. Dar pe o linie nouă,

se adaugă un punct cu un grad de neutilizare. Astfel, schimbarea netă a gradului total de neutilizare este de 1 punct la o mutare.
Deoarece o figură porneste cu n puncte, totalul gradelor de neutilizare initial este $3n$ si totalul acesta nu este niciodată negativ. Numărul de miscări este asadar finit.

b. Dati o margine superioară cât de bună se poate, pentru numărul de etape ale jocului Sprouts când începutul este cu n puncte si demonstrati rezultatul.

Solutie: Totalul gradelor de neutilizare poate atinge 1, dar niciodată 0 deoarece nu se poate trasa o linie dacă un singur grad neutilizat este disponibil. Astfel, limita superioară a celui mai îndelungat joc de Sprouts este 3n – 1.
Faptul poate fi demonstrat mai în deataliu cu invariantul următor. Fie propozitia $P(k)$ "după k miscări, numărul total al gradelor de neutilizare este de cel mult $3n - k$". Prin inductie se poate arăta usor că $P(k)$ este adevărată pentru orice k. Atunci $P(3n - 1)$ spune că după $3n - 1$ miscări rămâne un singur grad de neutilitate disponibil si astfel nu mai pot fi făcute alte miscări.

6. Demonstrati prin inductie că $\sum_{i=1}^{n} \lfloor i/2 \rfloor = \lfloor n^2/4 \rfloor$. Notatia $\lfloor x \rfloor$ este pentru cel mai mare întreg mai mic sau egal cu x.

Solutie: Se utilizează inductia tare. Fie $P(n)$ afirmatia că $\sum_{i=1}^{n} \lfloor i/2 \rfloor = \lfloor n^2/4 \rfloor$.

Cazuri de bază: Pentru $n = 1$ are loc evident $\lfloor 1/2 \rfloor = 0 = \lfloor 1^2/4 \rfloor$. Pentru $n = 2$, tot asa de evident $\lfloor 1/2 \rfloor + \lfloor 2/2 \rfloor = 0 + 1 = 1 = \lfloor 2^2/4 \rfloor$.

Pasul inductiv: Se admite adevărul pentru $P(k)$. Se urmăreste demonstrarea adevărului propozitiei $P(k + 1)$. Se consideră două cazuri:

(a) Dacă numărul k este par, atunci $k = 2m$, cu m potirivit ales. Atunci, folosind ipoteza inductivă

$$\sum_{i=1}^{k+1} \lfloor i/2 \rfloor = \sum_{i=1}^{k} \lfloor i/2 \rfloor + \lfloor (k+1)/2 \rfloor = \lfloor k^2/4 \rfloor + \lfloor (2m+1)/2 \rfloor =$$
$$= \lfloor 4m^2/4 \rfloor + \lfloor m + 1/2 \rfloor = m^2 + m$$

Similar

$$\lfloor (k+1)^2/4 \rfloor = \lfloor (4m^2 + 4m + 1)/4 \rfloor = \lfloor m^2 + m + 1/4 \rfloor = m^2 + m$$

Cocluzia: $P(k + 1)$ este adevărată pentru k par.

(a) Dacă numărul k este impar, atunci $k = 2m + 1$, cu m potirivit ales. Atunci, folosind ipoteza inductivă

$$\sum_{i=1}^{k+1} \lfloor i/2 \rfloor = \sum_{i=1}^{k} \lfloor i/2 \rfloor + \lfloor (k+1)/2 \rfloor = \lfloor k^2/4 \rfloor + \lfloor (2m+2)/2 \rfloor =$$
$$= \lfloor (4m^2 + 4m + 1)/4 \rfloor + m + 1 = m^2 + 2m + 1$$

Similar

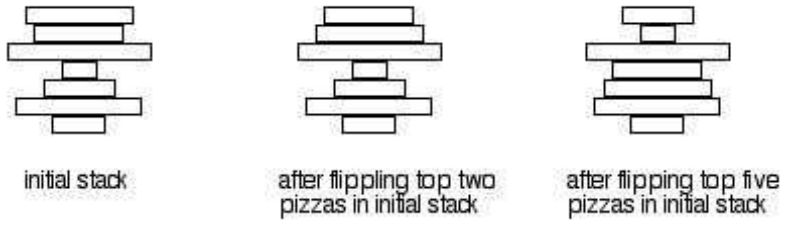
$$\lfloor (k+1)^2/4 \rfloor = \lfloor (4m^2 + 8m + 4)/4 \rfloor = m^2 + 2m + 1$$

Cocluzia: $P(k+1)$ este adevărată si pentru k impar.

De notat că demonstratia foloseste si adevărul $\lfloor m + x \rfloor = m + \lfloor x \rfloor$ ori de câte ori m este întreg.

7. **O demonstratie cu mai multe pizza**.

Lucrând la un local unde se serveste pizza, aveti o stivă de suporturi de pizza necoapte. Pentru o prezentare în formă nealterată, suporturile de pizza trebuie aranjate în ordinea dimensiunii: cea mai mare la bază, cea mai mică deasupra. Este posibil a pune spatula sub un teanc partial si să se rotească întregul teanc de deasupra spatulei, inversându-i ordinea. Figura alăturată arată rezultatele a două rotiri în genul celor descrise.



Acestea nu sunt singurele mutări posibile pentru a schimba ordinea stivei; în continuare, e de dorit a se repeta miscările acestea până se obtine o stivă ordonată.

Este totdeauna posibil a se ordona stiva! Demonstrati adevărul aceastei afirmatii.

Solutie: Da, stiva se poate ordona indiferent de numărul de suporturi de pizza în stivă. Demonstarea se face prin inductie.

Propozitia de demonstrat: $P(n)$ = "pornind de la orice stivă de n pizza, acestea se pot ordona folosind exclusiv rotirea unor stive (partiale)".

Cazul de bază: $P(1)$ este adevărată deoarece o stivă de o singură pizza este deja ordonată.

Pasul inductiv: a dovedi implicatia $P(n) \Rightarrow P(n+1)$ pentru orice $n \geq 1$.

(a) Ipoteza inductivă este că orice stivă de n pizza poate fi ordonată.

(b) De demonstrat: se poate ordona oricare stivă de $n + 1$ pizza.

(c) Se consideră o stivă oarecare de $n + 1$ pizza. Se localizează blatul cel mai mare, se asează spatula dedesubtul lui si se răstoarnă; această miscare produce o stivă de $n + 1$ pizza cu cea mai mare deasupra. Apoi se pune spatula sub întregul teanc si se răstoarnă din nou; rezultă o stivă de $n + 1$ pizza cu cea mai mare la bază. Acum se manevrează cele n pizza de deasupra celei situate la bază. Prin ipoteza inductivă, orice stivă de n blaturi poate fi ordonată prin răsturnări (repetate), în particular si stiva partială în discutie. Nici o miscare cu cele n blaturi nu va tulbura

pozitia blatului situat cel mai de jos, cel de al (n + 1)-lea. Astfel, deoarece stiva initială de n + 1 pizza a fost una arbitrară, orice stivă de n + 1 pizza poate fi ordonată dacă una de n poate fi ordonată.
Notă: Blatul cel mai mare poate să nu fie unic, dar trebuie să existe un blat care să fie cel putin la fel de mare ca oricare altul. Si în această situatie demonstratia rămâne validă.

- Ca alternativă, se poate pune propozitia $P(n)$ = "fiind dată o stivă de pizza nesortată, ea se poate aranja astfel ca cele mai mici pizza să fie în partea de sus, cele mai mari în partea de jos, ordonat".
- Cazul de bază: $P(1)$ este adevărată, evident adevărată: se poate pune spatula sub unicul blat si prin rotire se obine stiva ordonată.
- Pasul inductiv: $P(k) \Rightarrow P(k+1)$ pentru orice $1 < k \leq n$, unde n este numărul de pizza în stivă.

(a) Ipoteza inductivă spune că se poate sorta o stivă cu cele mai mici k pizza sus.
(b) De demonstrat: cea mai mică (k + 1) pizza poate fi pozitionată în vârful stivei.
(c) Odată ce cele mai mici k pizza au fost sortate, se pune spatula sub blatul cel mai mic de ordinul k + 1 si se răstoarnă stiva. Cele mai mici k pizza sunt acum în mijlocul stivei sortate cu cea mai mică la bază. Deasupra celor k pizza este un alt grup de pizza cu cea mai mică, a (k + 1)-a la vârf. Pentru răsturnarea următoare se pune spatula deasupra blatului k cel mai mic astfel că pizza (k + 1) cea mai mică este exact deasupra celei mai mici pizza k. În final, se pune spatula sub cea mai mică pizza si se răstoarnă încă o dată. Cea mai mică pizza (k + 1) este acum în vârf.

- Urmează concluzia: $P(n)$ este adevărată.

8. **Fii tu cel care dă note**.
Dă o notă de 10 sau de 4 fiecăreia din demonstratiile de mai jos. Dacă nota acordată este 4, explică exact relele din structura rationamentului pe care se bazează "demonstratia". Trebuie justificate toate explicatiile (a spune numai că "este fals" nu este o justificare).

- **Teorema 0.1**: Pentru orice $n \in \mathbf{N}$, $n^2 + n$ este impar.
Demonstratie: Demonstratia este prin inductie.
Cazul de bază: Numărul natural 1 este impar
Pasul inductiv: Se presupune $k \in \mathbf{N}$ si $k^2 + k$ este impar. Atunci
$(k+1)^2 + (k+1) = k^2 + 2k + 1 + k + 1 = (k^2 + k) + (2k + 2)$
este suma unui întreg impar cu unul par. În consecintă $(k+1)^2 + (k+1)$ este un număr impar. Pe baza principiului inductiei matematice, proprietatea că $n^2 + n$ este impar este adevărată pentru orice număr natural n.

□

Solutie: Aici se acordă un 4. Cazul de bază este invalid deoarece nu se potriveste cu ceea ce se cere demonstrat.

- **Teorema 0.2**: Pentru orice x, $n \in \mathbf{N}$, dacă $nx = 0$ si $n > 0$, atunci $x = 0$.
  Demonstratie: Demonstratia este prin inductie.
  Cazul de bază: Dacă $n = 1$ atunci ecuatia $nx = 0$ implică $nx = 1.x = x$ în acest caz.
  Pasul inductiv: Fixăm $k > 0$ si presupunem că are loc $kx = 0$ care implică $x = 0$. Să presupunem că are loc $(k + 1)x = 0$. Se observă relatia $(k + 1)x = kx + x$, asadar se poate conchide că are loc $kx + x = 0$, cu alte cuvinte $kx = – x$. Sunt două cazuri posibile:
  Cazul 1: $x = 0$. În acest caz, $kx = – x = – 0 = 0$, astfel încât $kx = 0$. În consecintă, ipoteza inductivă spune că $x = 0$.
  Cazul 2: $x > 0$. În acest caz, $– x < 0$ (deoarece $x > 0$). În acelasi timp, $kx \geq 0$ (deoarece k, $x \geq 0$). Dar asta este imposibil, deoarece se stie că are loc $kx = – x$. Avem o contradictie si de aceea, cazul 2 nu poate apărea.
  În ambele cazuri, se poate conchide că $x = 0$. Asta încheie demonstrarea pasului inductiv.

□

Solutie: Demonstratia primeste un 10. (Sunt căi mai simple de a demonstra acelasi lucru, dar demonstratia este corectă).

- **Teorema 0.3**: Pentru orice x, y, $n \in \mathbf{N}$, dacă $\max(x, y) = n$, atunci $x = y$.
  Demonstratie: Demonstratia este prin inductie.
  Cazul de bază: Se presupune că $n = 0$. Dacă $\max(x, y) = 0$ si x, $y \in \mathbf{N}$, atunci $x = 0$ si $y = 0$ si $x = y$.
  Pasul inductiv: Se presupune că ori de câte ori $\max(x, y) = k$, rezultă $x = y$. Mai departe pesupunem că x si y sunt astfel încât $\max(x, y) = k + 1$. Urmează din asta că $\max(x – 1, y – 1) = k$ si prin ipoteza inductivă $x – 1 = y – 1$. Rezultă $x = y$, ceea ce completează pasul inductiv.

□

Solutie: Se acordă un 4. Pasul inductiv este slab formulat: ipoteza inductivă nu este aplicată corect. Este adevărat că $\max(x – 1, y – 1) = k$ dar ipoteza inductivă nu se aplică în mod necesar aici deoarece $x – 1$ sau $y – 1$ ar putea fi negative si ipoteza inductivă se aplică numai când argumentele functiei "max" sunt numere naturale. Ca rezultat, nu putem conchide în sigurantă că $x – 1 = y – 1$.
De pildă, se poate vedea ce merge rău dacă se încearcă a pune $k = 0$, $x = 0$, $y = 1$. Este atunci adevărat că $\max(x, y) = k + 1$ (deoarece $\max(0, 1) = 1$) si este adevărat că $\max(x – 1, y – 1) = k$ (deoarece $\max(–1, 0) = 0$) dar nu este adevărat că $x – 1 = y – 1$ (ipoteza inductivă garantează numai că dacă $\max(0, 0) = 0$ atunci $0 = 0$, dar nu promite nimic în cazul argumentelor din "max" negative).

Demonstratia ar putea fi făcută mai putin confuză dacă autorul ar fi declarat explicit faptul că inductia se face după k. Cu toate acestea, aceasta nu este o eroare în demonstratie: o scădere a aprecierii de la 10 la 8 este suficientă.

- **Teorema 0.4**: $\forall\ n \in \mathbf{N}$, $n^2 \leq n$.
  Demonstratie: Demonstratia este prin inductie.
  Cazul de bază: Dacă $n = 0$, afirmatia $0^2 \leq 0$ este adevărată.
  Pasul inductiv: Se presupune $k \in \mathbf{N}$ si $k^2 \leq k$. Trebuie arătat că $(k + 1)^2 \leq k + 1$.
  Lucrând în sens invers se vede că:

$$(k + 1)^2 \leq k + 1$$
$$k^2 + 2k + 1 \leq k + 1$$
$$k^2 + 2k \leq k$$
$$k^2 \leq k$$

  Astfel se revine la ipoteza initială presupusă a fi adevărată. Asadar, pentru orice $n \in \mathbf{N}$ stim că $n^2 \leq n$.

□

Solutie: Rationamentul în această demonstratie este inversat, admis ca atare. În consecintă, este un caz clasic de eroare prin inversare. Se acordă un 4.

1. Numerele Fibonacci

Numerele Fibonacci sunt definite astfel:

$F_0 = 0$

$F_1 = 1$

$F_n = F_{n-2} + F_{n-1}$ pentru $n > 1$.

(a) Listati primele zece numere Fibonacci (F_0 la F_9)

Solutie:

n	0	1	2	3	4	5	6	7	8	9
F_n	0	1	1	2	3	5	8	13	21	34

(b) Se consideră următoarea procedură Scheme, care fiind dat n generează F_n

```
(define (fib n)
  (cond
      ((= n 0) 0)
      ((= n 1) 1)
      (else (+ (fib (- n 1)) (fib (- n 2)))) ) )
```

Fie T_n numărul de operatii de adunare necesare în calculul lui `(fib n)`. Listati valorile T_0 până la T_9.

Solutie:

n	0	1	2	3	4	5	6	7	8	9
T_n	0	0	1	2	4	7	12	20	33	54

(c) Formulati si demonstrati o relatie între numerele T si numerele F.

Solutie: $T_n = F_{n+1} - 1$ pentru orice $n \in$ N.

Demonstratie: se face prin inductia tare după n.

Cazurile de bază $n = 0$ si $n = 1$ se verifică deoarece nici unul din apelurile `(fib 0)` si `fib 1)` nu este recursiv la `fib`. Astfel, $T_0 = 0 = F_1 - 1$ si $T_1 = 0 = F_2 - 1$.

Dacă $n \geq 2$, `(fib n)` se evaluează prin adunarea lui `(fib (- n 1))` cu `(fib (- n 2))`, fiecare necesitând respectiv T_{n-1} si T_{n-2} adunări. Acest fapt produce relatia de recurentă

$$T_n = T_{n-1} + T_{n-2} + 1$$

Pasul inductiv: Se presupune egalitatea $T_n = F_{n+1} - 1$ adevărată pentru orice număr natural $n \leq k$. Dorim să arătăm că $T_{k+1} = F_{k+2} - 1$. Argumentatia continuă astfel:

$T_{k+1} = T_k + T_{k-1} + 1 = (F_{k+1} - 1) + (F_k - 1) + 1 = F_{k+1} + F_k - 1 = F_{k+2} - 1$

Secventa de egalităti ia în calcul relatia de recurentă de mai sus, ipoteza inductivă si definitia numerelor Fibonacci.

(d) Demonstrati că $F_{n+1} F_{n-1} - F_n^2 = (-1)^n$.

Solutie: Se utilizează inductia după n. Cazul de bază, pentru $n = 1$ are loc deoarece

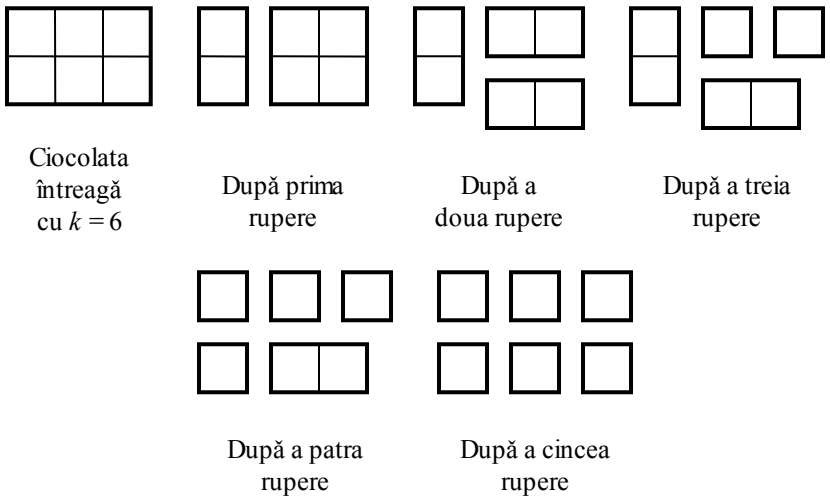
$$F_2F_0 - F_1^2 = 1\times0 - 1^2 = 0 - 1 = -1 = (-1)^1$$

Pentru pasul inductiv, se arată că dacă identitatea are loc pentru $n = k$, atunci ea se mentine si pentru $n = k + 1$.

$$F_{k+2}F_k - F_{k+1}^2 = (F_{k+1} + F_k)\,F_k - F_{k+1}^2 = F_{k+1}F_k + F_k^2 - F_{k+1}^2 =$$
$$= F_{k+1}(F_k - F_{k+1}) + F_k^2 = -F_{k+1}F_{k-1} + F_k^2 = -(-1)^k = (-1)^{k+1}$$

2. Împărtirea ciocolatei

Ciocolata se prezintă adesea ca un dreptunghi împărtit în pătrate (dreptunghiuri) mai mici. Este la îndemâna oricui să rupă un dreptunghi mai mare în două mai mici de-a lungul liniilor longitudinale sau transversale care delimitează micile pătrate. Figura alăturată este un exemplu.



Se presupune că ciocolata este alcătuită din k pătrate si trebuie măruntită până la pătratele elementare. Demonstrati că indiferent de ordinea în care este fragmentată, tot de exact $k - 1$ actiuni de rupere este nevoie.

Solutie: Se utilizează inductia tare după k. Evident, pentru cazul de bază nu este necesară vreo rupere pentru o ciocolată dintr-o singură bucată ($k = 1$). Pentru pasul inductiv, se admite că pentru orice $k \le n$ sunt necesare exact $k - 1$ ruperi pentru o ciocolată din k bucăti. Pentru o ciocolată din $k + 1$ bucăti secventa de ruperi începe cu obtinerea unei bucăti de x pătrătele si a unei bucăti de $k + 1 - x$ pătrătele cu $x \in$ N si $0 < x < k + 1$. Conform ipotezei inductive, aceste bucăti mai mici vor necesita respectiv exact $x - 1$ si exact $k - x$ operatii de rupere. Astfel că pentru portionarea ciocolatei alcătuite din $k + 1$ pătrătele sunt necesare $(x - 1) + (k - x) + 1 = k$ ruperi.

3. Ghicitorul

Un student la masterat în AII, încearcă un joc de-a ghicitul si scrie următorul pseudocod.

tipăreste "Gânditi-vă la un număr între 1 si 100 si eu îl voi ghici"
celmaimicposibil = 1;
celmaimareposibil = 100;
atât timp cât (nu s-a ghicit numărul)
    $miloclul = \lfloor (celmaimicposibil + celmaimareposibil)/2 \rfloor$;
    print "Este ", *mijlociul*, "?";
    dacă chestionatul spune da, s-a ghicit numărul
    alminteri
      tipăreste "Este mai mare decât ", *mijlociul*, "?";
      dacă chestionatul spune da, pune *celmaimicposibil* = *mijlociul*
      altminteri pune *celmaimareposibil* = *mijlociul*;

(a) Studentul intentionează să mentină următoarea buclă invariantă: numărul de ghicit este undeva în intervalul [*celmaimicposibil*, *celmaimareposibil*]. Este acest invariant mentinut la fiecare buclă parcursă? Explicati de ce da sau de ce nu.

Solutie: Da. Fie *N* numărul de ghicit. Se porneste cu *N* în intervalul
[*celmaimicposibil*, *celmaimareposibil*]
Dacă *N* > *mijlociul*, atunci *N* este în intervalul [*mijlociul*, *celmaimareposibil*]. Dacă *N* < *mijlociul*, atunci *N* este în intervalul [*celmaimicposibil*, *mijlociul*]. Asadar, în ambele cazuri intervalul [*celmaimicposibil*, *celmaimareposibil*] este actualizat corespunzător.

(b) Listati toate numerele dintre 1 si 100, inclusiv, pe care algoritmul nu le poate ghici si descrieti ce se întâmplă cu fiecare.

Solutie: Algoritmul esuează în a ghici numărul 100. Intervalele sunt actualizate după cum urmează: [1, 100], [50, 100], [75, 100], [87, 100], [93, 100], [96, 100], [98, 100], [99, 100]. După această actualizare algoritmul "ghiceste" fără încetare 99.

(c) Presupunem că facem două schimbări în algoritm: prima, se initalizează *celmaimareposibil* la 101; a doua, se actualizează *celmaimicposibil* la *mijlociul* + 1 dacă valoarea celui chestionat este mai mare decât *mijlociul*. Codul modificat mentine o buclă invariantă care este mai tare decât aceea descrisă la punctul (a). Descrieti bucla invariantă din codul modificat si explicati răspunsul.

Solutie: Numărul *N* de ghicit este în intervalul [*celmaimicposibil*, *celmaimareposibil* – 1]. Acest invariant este adevărat după setarea initială deoarece intervalul este [1, 101 – 1] = [1, 100]. După fiecare ghicire incorectă intervalul este actualizat corespunzător astfel că *mijlociul* este exclus.

(d) Utilizând invariantul actualizat, formulati o demonstratie prin inductie pe care algoritmul modificat lucrează corect.

Solutie: Fie N numărul de ghicit si fie prin definitie *diferenta* = *celmaimareposibil* – *celmaimicposibil*. De observat mai întâi că dacă *diferenta* = 1 algoritmul se încheie deoarece *mijlociul* atinge singura valoare rămasă accesibilă, respectiv *celmaimicposibil*.

Arătăm că atât timp cât *diferenta* > 1, în următoarea buclă iterativă fie *diferenta* se micsorează, fie algoritmul se încheie. Sunt de examinat trei cazuri:

i. Dacă *mijlociul* este răspunsul corect, algoritmul se încheie.

ii. Dacă *diferenta* este un număr par $2k$, atunci *mijlociul* = *celmaimicposibil* + k. Dacă N > *mijlociul*, noua valoare pentru *diferenta* devine

$$celmaimareposibil - (celmaimicposibil + k + 1) = 2k - k - 1 = k - 1.$$

Dacă N < *mijlociul*, atunci *diferenta* devine

$$(celmaimicposibil + k) - celmaimicposibil = k.$$

Dar k si $k - 1$ sunt mai mici decât $2k$, astfel că *diferenta* descreste.

iii. Dacă *diferenta* este un număr impar $2k + 1$, atunci *mijlociul* = *celmaimicposibil* + k. În ambele cazuri, N > *mijlociul* sau N < *mijlociul*, noua valoare pentru *diferenta* devine k.

Asadar, *diferenta* descreste, dar rămâne încă pozitivă. Astfel *diferenta* trebuie să atingă în cele din urmă 1 si algoritmul se încheie cu valoarea corectă.

4. Calcul max-min

Fiind dată o multime de N valori numerice, se poate găsi un element maxim prin $N - 1$ comparatii si apoi un element minim în alte $N - 2$ comparatii (omitând elementul maxim), în total $2N - 3$ comparatii. Se consideră algoritmul următor:

dacă multimea are două elemente, atunci
    cu o comparatie se determină cel mai mare si cel mai mic element
      si se memorează în variabilele *maimare* si *maimic*;
    se returnează perechea [*maimare*, *maimic*];
altminteri
        se împarte multimea de elemente în jumătăti;
    se face un apel recursiv pentru a găsi cel mai mare si cel mai mic element din prima jumătate;
    se face un apel recursiv pentru a găsi cel mai mare si cel mai mic element din a doua jumătate;
    se pune *maimare* la valoarea celui mai mare dintre cele două elemente cele mai mari (cu o comparatie);
    se pune *maimic* la valoarea celui mai mic dintre cele două elemente cele mai mici (cu o comparatie);
    se returnează perechea [*maximum*, *minimum*];

(a) Demonstrati că, pentru N o putere a lui 2, algoritmul de mai sus necesită totdeauna mai putin de $3N/2$ comparatii (ignorând comparatiile necesare eventual pentru a împărti multimile în două).

Solutie: Se demonstrează o propozitie mai tare $P(N)$: algoritmul execută $3N/2 - 2$ comparatii pentru o listă de N elemente. Demonstratia se face prin inductie. Cazul de bază $P(2)$ este adevărat: numai o comparatie se execută în acest caz si $1 = (3.2)/2 - 2$. Pentru pasul inductiv trebuie demonstrat că $P(N) \Rightarrow P(2N)$. Pe o listă de $2N$ elemente, algoritmul găseste recursiv *minimum* si *maximum* din lista initială. Fie $C(N)$ numărul de comparatii efectuat pe lista de întindere N. Atunci

$$C(2N) = 2C(N) + 2 = 2(3N/2 - 2) + 2 = 3N - 2 = 3(2N)/2 - 2$$

asa încât

$$C(N) = 3N/2 - 2$$

pentru orice N care este o putere a lui 2.

(b) Identificati ratiunea pentru care acest algoritm lucrează mai rapid decât metoda directă descrisă mai devreme.

Solutie: Dacă se evaluează numai valoarea minimă cu algoritmul recursiv, sunt necesare numai $N - 1$ comparatii. Astfel, pentru a stabili *minimum* si *maximum* separat s-ar "consuma" $2N - 2$ comparatii care reprezintă grosier numărul de comparatii executate cu metoda diectă. Prin stabilirea concomitentă a celor două valori, se economisesc $N/2$ comparatii deoarece fiecare pereche de numere suferă la nivelul cel mai de jos al schemei recursive numai o comparatie si nu două (cum s-ar întâmpla în cazul separat). De observat că nu numai recursivitatea este "vinovată" de această accelerare.

5. Analiza frunzelor

Un arbore binar cu rădăcină este fie vid, fie constă dintr-o rădăcină si zero, unul sau doi subarbori disjuncti (care sunt si ei arbori binari cu rădăcină). O frunză este un arbore binar cu rădăcină cu o rădăcină si fără subarbori.

(a) Demonstrati că, într-un arbore binar cu rădăcină si cu L frunze

$$\sum_{k=1}^{L} 2^{-d(k)} \leq 1$$

unde $d(k)$ este adâncimea frunzei k. Adâncimea rădăcinii unui arbore este 0 si adâncimea oricărui alt nod este 1 plus adâncimea părintelui său.

Solutie: Se defineste pe un arbore T functia

$$f(T) = \sum_{k=1}^{L(T)} 2^{-d_T(k)}$$

Suma se întinde pe frunzele arborelui T. Aici $L(T)$ exprimă numărul frunzelor arborelui T si $d_T(k)$ exprimă adâncimea frunzei k în T (numărul de pasi pe calea de la rădăcină la frunza k). Se dau două demonstratii că $f(T) \leq 1$ pentru orice arbore T: una prin inductie structurală, alta prin inductie după adâncimea arborelui.

Demonstratie (prin inductie structurală):

Cazuri de bază: Dacă T este un arbore vid, atunci $f(T) = 0 \leq 1$. Dacă T este un arbore cu rădăcină fără descendenti, atunci $f(T) = 2 - 0 = 1 \leq 1$.
Pasul inductiv partea #1: Este necesar a arăta că pentru orice arbore T_0, dacă T este arborele a cărui rădăcină are pe T_0 ca unic descendent, atunci $f(T_0) \leq 1 \Rightarrow f(T) \leq 1$. Fie T_0 arbitrar si să admitem că $f(T_0) \leq 1$. De observat că frunzele lui T sunt aceleasi cu cele ale lui T_0 cu adaosul că adâncimea lor a crescut cu o unitate. Cu alte cuvinte, $L(T) = L(T_0)$ si $d_T(k) = 1 + d_{T0}(k)$ pentru orice $1 \leq k \leq L(T)$. Asadar

$$f(T) = \sum_{k=1}^{L(T)} 2^{-d_T(k)} = \sum_{k=1}^{L(T_0)} 2^{-1-d_{T_0}(k)} = \frac{1}{2}\sum_{k=1}^{L(T_0)} 2^{-d_{T0}(k)} = \frac{1}{2} f(T_0) \leq \frac{1}{2} \leq 1$$

Pasul inductiv partea #2: Trebuie arătat că pentru orice pereche de arbori T_0, T_1, dacă T este arborele a cărui rădăcină are pe T_0 si pe T_1 ca descendenti, atunci $(f(T_0) \leq 1 \wedge f(T_1) \leq 1) \Rightarrow f(T) \leq 1$. Fie T_0, T_1 arbitrari si să presupunem că $f(T_0) \leq 1$ si $f(T_1) \leq 1$. De observat că frunzele lui T sunt reuniunea frunzelor lui T_0 si T_1, doar că adâncimea lor a crescut cu o unitate. Cu alte cuvinte, $L(T) = L(T_0) + L(T_1)$, $d_T(k) = 1 + d_{T0}(k)$ pentru orice $1 \leq k \leq L(T_0)$ si $d_T(k) = 1 + d_{T1}(k - L(T_0))$ pentru orice $L(T_0) + 1 \leq k \leq L(T_0) + L(T_1)$. Asadar

$$f(T) = \sum_{k=1}^{L(T)} 2^{-d_T(k)} = \sum_{k=1}^{L(T_0)} 2^{-1-d_{T_0}(k)} + \sum_{k=1}^{L(T_1)} 2^{-1-d_{T_1}(k)} =$$

$$= \frac{1}{2}\sum_{k=1}^{L(T_0)} 2^{-d_{T_0}(k)} + \frac{1}{2}\sum_{k=1}^{L(T_1)} 2^{-d_{T_1}(k)} = \frac{1}{2} f(T_0) + \frac{1}{2} f(T_1) \leq \frac{1}{2} + \frac{1}{2} \leq 1$$

□
Demonstratie (prin inductie după adâncimea arborelui, adâncimea ultimei frunze din arbore):
Pentru cazul de bază $d = 0$, singura frunză din T este rădăcina. Se verifică imediat că $f(T) = 2^{-0} \leq 1$.
Înaintea demonstrării pasului inductiv se introduce o lemă: Fie T un arbore. Se defineste arborele T' în care rădăcina lui T este subarborele din stânga în T' si T' nu are un subarbore în dreapta.
Lemă: $f(T') = (1/2)f(T)$.
Demonstratie (a lemei): Fiecare frunză din T' este o frunză în T. Adâncimea $d'(k)$ a frunzei k din T' este 1 plus adâncimea $d(k)$ a acelei frunze în T. Astfel

$$f(T') = \sum_{k=1}^{L} 2^{-d'(k)} = \sum_{k=1}^{L} 2^{-[d(k)+1]} = \frac{1}{2}\sum_{k=1}^{L} 2^{-d(k)}$$

ceea ce completează demonstratia.
Pasul inductiv: Ca ipoteză inductivă, se presupune că $f(T') \leq 1$ se mentine pentru toti arborii T' de adâncime cel mult d. Se consideră un arbore T cu adâncimea $(d + 1)$. T are cel putin un subarbore T_1 de adâncime d. Prin ipoteza inductivă, $f(T_1) \leq 1$. Dacă T_1 este unicul subarbore, atunci

$$f(T) = (1/2)f(T_1) \leq (1/2) < 1$$

Dacă T are un al doilea subarbore T_2, adâncimea lui T_2 este cel mult d. Din nou, prin ipoteza inductivă, $f(T_2) \leq 1$. În acest caz

$$f(T) = (1/2)f(T_1) + (1/2)f(T_2) \leq (1/2) + (1/2) = 1$$

□

(b) Descrieti complet arborii pentru care

$$\sum_{k=1}^{L} 2^{-d(k)} = 1$$

si demonstrati afirmatiile pe care le faceti. Asta va insemna două demonstratii: una pentru verificarea faptului că arborii descrisi satisfac egalitatea, alta pentru a arăta că toti ceilalti arbori dau o sumă mai mică decât unitatea.

Solutie: T este un *arbore binar complet* dacă fiecare nod al lui are 0 sau 2 descendenti. Se demonstrează că T este un arbore binar complet dacă si numai dacă

$$f(T) = \sum_{k=1}^{L} 2^{-d(k)} = 1$$

Demonstratie: Se foloseste inductia tare pe adâncimea arborilor.

Pentru cazul de bază, $d = 0$: un arbore constând numai din rădăcină nu are descendenti si $f(T) = 2^0 = 1$.

Pentru pasul inductiv, se admite că afirmatia este valabilă pentru toti arborii de adâncime cel mult d si se demonstrează că afirmatia este adevărată pentru arbori de adâncime $(d + 1)$. Fie T un arbore de adâncime $(d + 1)$ si admitem mai întâi că T este binar complet. Rădăcina lui T trebuie să aibă doi descendenti, astfel că subarborii T_1 si T_2 au fiecare adâncimi de cel mult d. Atât T_1 cât si T_2 sunt arbori binari completi asa încât $f(T_1) = 1$ si $f(T_2) = 1$. Prin acelasi rationament parcurs la punctul (a) al problemei

$$f(T) = (1/2)f(T_1) + (1/2)f(T_2) = (1/2) + (1/2) = 1$$

Asadar, orice arbore binar complet T de adâncime $(d + 1)$ satisface relatia $f(T) = 1$.

Se consideră acum un arbore binar T de adâncime $(d + 1)$ care nu este complet. În acest caz, unul din noduri trebuie să aibă exact un descendent. Dacă rădăcina lui T are numai un descendent, atunci T are numai un subarbore T_1, astfel că

$$f(T) = (1/2)f(T_1) \leq 1/2 < 1$$

Dacă T are doi subarbori T_1 si T_2, atunci cel putin unul din ei are unul din noduri cu un singur descendent. Fără pierdere de generalitate se poate admite că T_1 este acel subarbore. Prin ipoteza inductivă, $f(T_1) < 1$. Atunci

$$f(T) = (1/2)f(T_1) + (1/2)f(T_2) < (1/2) + (1/2) = 1$$

Asadar, nici un arbore binar T incomplet de adâncime $(d + 1)$ nu poate satisface egalitatea $f(T) = 1$. Aceasta încheie demonstrarea pasului inductiv si demonstratia prin inductie.

□

De observat că demonstratiile inductive pentru ambele părti ar putea fi bazate pe numărul de noduri dintr-un arbore si nu pe adâncimea acelui arbore.

1. Arbori de decizie booleeni

Un *arbore decizional* este o variantă a arborilor binari în care nodurile *interne* sunt etichetate cu o variabilă booleană. Fiecare frunză este un atom T sau F. Notatia $(P, t_1 \bullet t_2)$ este pentru un arbore decizional a cărui rădăcină este etichetată cu P si ai cărui subarbori la stânga si la dreapta sunt t_1 si t_2. Un arbore decizional t reprezintă o functie booleană după cum urmează, cu m un model (adică o asignare prin care se atribuie fiecărei variabile booleene valori *true* sau *false*):

dacă t este un atom, $eval(t, m) = t$;
$eval((P, t_1 \bullet t_2), m) = eval(t_1, m)$ dacă P este în m *true*
$eval((P, t_1 \bullet t_2), m) = eval(t_2, m)$ dacă P este în m *false*

(a) Desenati un arbore decizional care reprezintă aceeasi functie booleană ca si $A \Rightarrow B$.

Solutie: $(A, (B, T \bullet F) \bullet T)$

(b) Cât de extins este arborele decizional cel mai mic (adică arborele cu număr minim de noduri) care reprezintă disjunctia a n variabile? Justificati pe scurt răspunsul.

Solutie: Cel mai redus arbore decizional are $2n + 1$ noduri. Fiecare variabilă a unui nod interior si fiecare subarbore la stânga al acestor noduri este atomul simplu T. Nodurile interioare sunt legate ca un lant de-a lungul ramurii cea mai din dreapta care se termină cu atomul F.

De ce nu poate fi încă micsorat? Functia booleană $f(X_1, \ldots, X_n) = X_1 \vee \ldots \vee X_n$ depinde toate cele n variabile booleene ale sale. În consecintă, fiecare din acele variabile trebuie să apară separat într-unul dintre nodurile interioare ale arborelui. Deoarece sunt n variabile, orice arbore de acest gen va trebui să aibă cel putin n noduri interioare. Mai mult, orice arbore binar complet cu n noduri interioare trebuie să aibă cel putin $n + 1$ frunze (cum poate fi arătat prin inductie), asadar orice arbore care reprezintă functia aceasta trebuie să aibă în total cel putin $2n + 1$ noduri.

(c) Demonstrati că oriec functie booleană poate fi reprezentată ca un arbore decizional.

Solutie: Se foloseste inductia după numărul n de variabile dintr-o formulă. Când este în joc numai o variabilă booleană X_1 în functia $f(X_1)$, arborele de decizie este dat de $(X_1, f(T) \bullet f(F))$. Cu alte cuvinte, frunzele sunt valorile obtinute prin evaluarea functiei f în $X_1 = T$ si în $X_1 = F$.

Pentru pasul inductiv, se presupune că orice functie booleană cu cel mult k variabile poate fi reprezentată ca un arbore de decizie. Fie $f(X_0, \ldots, X_k)$ o functie booleană cu $k + 1$ variabile. Se construieste un arbore decizional pentru f după cum urmează. Se definesc functiile booleene f_1 si f_2, $f_1(X_1, \ldots, X_k) = f(T, X_1, \ldots, X_k)$ si $f_2(X_1, \ldots, X_k) = f(F, X_1, \ldots, X_k)$. Ambele sunt functii

de k variabile si sunt reprezentabile conform ipotezei inductive prin arborii t_1 si t_2. În consecintă, functia f de $k + 1$ variabile booleene poate fi reprezentată prin arborele $(X_0, t_1 \bullet t_2)$.

(d) Desenati un arbore decizional care reprezintă functia de X_1, X_2, X_3 următoare: T dacă cel putin două din cele trei variabile sunt *true*, altminteri F.

Solutie: $(X_1, (X_2, T \bullet (X_3, T \bullet F)) \bullet (X_2, (X_3, T \bullet F) \bullet F))$

(e) Scrieti o definitie matematică recursivă pentru functia $kn(k, n)$ care returnează un arbore decizional pe n variabile $X_1, \ldots, X_n$; arborele de decizie returnează T dacă si numai dacă cel putin k din cele n variabile sunt *true*.

Solutie:

$$kn(k, n) = \begin{cases} T & \text{pentru } k = 0 \\ F & \text{pentru } k > n \\ (X_n, kn(k-1, n-1) \bullet kn(k, n-1)) & \text{altminteri} \end{cases}$$

2. O problemă de orar

Se consideră următoarea problemă de programare a examenelor. Sunt n candidati, $c_1, c_2, \ldots, c_n$; m intervale de timp disjuncte, $t_1, t_2, \ldots, t_m$ si l examene $e_1, e_2, \ldots, e_l$. Pentru fiecare candidat se dă $E(c_i)$, submultimea examenelor pe care candidatul trebuie să le sustină. Un *plan* este orice atribuire a examenelor la intervalele de timp. Un plan este *fezabil* dacă nici un candidat nu trebuie să fie prezent în acelasi timp la două examene. Se caută un plan fezabil. De notat că problema este interesantă numai dacă $l > m$, adică dacă numărul de exameme depăseste numărul de intervale de timp. Altminteri, se planifică examenele în intervale disjuncte de timp si nu vor fi suprapuneri de nici un fel.

Să vedem cum se asează această problemă în cadrul unui rationament logic, prin construirea unei expresii booleene S care poate fi satisfăcută dacă si numai dacă există un plan fezabil. Evident, S depinde de datele problemei (adică de multimea particulară de candidati, ca si de n, m, l). Variabilele din formula lui S vor fi X_{ij}, una pentru fiecare examen e_i si fiecare interval de timp t_j, unde X_{ij} este *true* dacă si numai dacă examenul e_i este planificat în intervalul t_j.

(a) Arătati mai întâi cum se scrie o formulă S' care exprimă numai faptul că X_{ij} codează un plan propriu, complet (ignorând fezabilitatea din punct de vedere al studentilor). Se poate utiliza oricare dintre sintaxele standard pentru expresii booleene, inclusiv operatorii $\wedge$, $\vee$, $\neg$, $\Rightarrow$. Se poate utiliza si notatii sintetice precum $\wedge_{i=1}^{n} Y_i$ pentru o conjunctie multiplă, similar cu $\prod_{i=1}^{n} y_i$ pentru un produs de mai multe numere. (*Indicatie*: Ocupati-vă de fiecare examen separat. Ce trebuie făcut cu fiecare examen?).

Solutie: Trebuie exprimat faptul că fiecare examen este planificat cel putin într-un interval de timp si că un examen e_i odată planificat în intervalul t_j, nu mai este planificat în alt interval de timp.

$$S' = \wedge_{i=1}^{l} \left(\vee_{j=1}^{m} X_{ij} \wedge \wedge_{j=1}^{m} (X_{ij} \Rightarrow \wedge_{k=1, k \neq j}^{m} \neg X_{ik}) \right)$$

(b) Arătati acum cum se adaugă restrictii suplimentare care cer ca planul să fie fezabil. Apoi deduceti cum se construieste întreaga formulă *S*.

Solutie: Trebuie exprimat faptul că niciodată două examene sustinute de un student nu pot fi în acelasi interval de timp. Sunt două posibiltăti de a exprima acest lucru:

$$S = S' \wedge \wedge_{h=1}^{n} \wedge_{i=1}^{l} \left(e_i \in E(c_k) \Rightarrow \wedge_{j=1}^{m} (X_{ij} \Rightarrow \wedge_{k=1, k \neq i}^{l} (e_k \in E(c_n) \Rightarrow \neg X_{kj})) \right)$$

$$S = S' \wedge \wedge_{h=1}^{n} \wedge_{i=1}^{l-1} \wedge_{k=i+1}^{l} \left((e_i \in E(c_h) \wedge e_k \in E(c_h)) \Rightarrow \wedge_{j=1}^{m} \neg (X_{ij} \wedge X_{kj}) \right)$$

(c) Verificati formula stabilită prin arătarea (i) modului cum se construieste o atribuire care satisface pe *S* dat fiind un plan fezabil oarecare si (ii) modul cum se construieste un plan fezabil fiind dată o atribuire care satisface pe *S*.

Solutie: Fiind dat un plan fezabil, se construieste o atribuire care satisface pe *S* prin punerea variabilelor X_{ij} pe valoarea "true" dacă examenul e_i este atribuit intervalului de timp t_j si pe "false" în alte cazuri. Fiind dată o atribuire care satisface pe *S*, se crează un plan fezabil prin atribuirea examenului e_i intervalului de timp t_j dacă si numai dacă X_{ij} este "true".

(d) Există o bijectie (o corespondentă biunivocă deplină) între atribuirile care satisfac si planurile fezabile? Sustineti pe scurt răspunsul dat.

Solutie: Da. A se utiliza constructia de la punctul (c).

3. Formularea unei probleme de Minesweeper

Se consideră următoarea problemă de Minesweeper:

2			
1	1	2	1
	1	2	3

(a) Listati expresiile CNF care corespund restrictiilor locale $N_{1,1}$, $N_{2,1}$, $N_{3,1}$ rezultate din pătratele (1, 1), (2, 1), (3, 1).

Solutie: Fie $X_{i,j}$ variabla cu semnificatia "există o mină în pătratul (i, j)".

$N_{1,1} = (X_{1,2} \vee X_{2,2}) \wedge (\neg X_{1,2} \vee \neg X_{2,2})$

$N_{2,1} = (X_{1,2} \vee X_{2,2}) \wedge (X_{1,2} \vee X_{3,2}) \wedge (X_{2,2} \vee X_{3,2}) \wedge (\neg X_{1,2} \vee \neg X_{2,2} \vee \neg X_{3,2})$

$N_{3,1} = (X_{2,2} \vee X_{3,2}) \wedge (\neg X_{2,2} \vee \neg X_{3,2})$

(b) Construiti o tabelă de adevăr pentru problemă, care trebuie să aibă 8 linii. Adăugati coloane pentru expresiile $N_{1,1}$, $N_{2,1}$, $N_{3,1}$.

Solutie:

$X_{1,2}$	$X_{2,2}$	$X_{3,2}$	$N_{1,1}$	$N_{2,1}$	$N_{3,1}$
T	*T*	*T*	*F*	*F*	*F*
T	*T*	*F*	*F*	*T*	*T*
T	*F*	*T*	*T*	*T*	*T*
T	*F*	*F*	*T*	*F*	*F*
F	*T*	*T*	*T*	*T*	*F*
F	*T*	*F*	*T*	*F*	*T*
F	*F*	*T*	*F*	*F*	*T*
F	*F*	*F*	*F*	*F*	*F*

(c) Marcati acele linii care corespund atribuirilor care satisfac acest set de restrictii.

Solutie: A treia linie ($X_{1,2}$ = *T*, $X_{2,2}$ = *F*, $X_{3,2}$ = *T*) este o atribuire care satisface.

(d) Deduceti ce puteti deduce relativ la pătratele necunoscute.

Solutie: Pătratele (1, 2) si (3,2) au mine, (2, 2) este neminat.

1. **Admitere la facultate**. Se consideră problema admiterii a n studenti la k specializări. Specializarea j poate admite cel mult n_j studenti. Fiecare student are o ordine a preferintelor relativ la specializări si fiecare specializare are o ordine a preferintelor relativ la studenti. O *admitere stabilă* este considerată o atribuire de studenti pe specializări fără "cupluri nesincere" adică fără situatii în care să existe un (o) student(ă) A care nu este admis(ă) la specializarea a, dar el (ea) preferă a si nu specializarea lui/ei si este preferat(ă) de a cel putin unuia din studentii din a. Descrieti cum se poate utiliza algoritmul traditional al împerecherii (TMA – Traditional Matching Algorithm[1]) ca o subrutină, pentru a produce o admitere stabilă (codul asociat algortimului trebuie modificat!). Justificati de ce această procedură produce o admitere stabilă.

Solutie: Problema împerecherii se poate adapta astfel ca studentii să fie potriviti cu *locurile de admitere* (*admission slots*). Mai întâi, să ne asigurăm că numărul de locuri de admitere este egal cu cel al studentilor prin adăugarea de locuri imaginare sau de studenti imaginari. Colegiul j are n_j locuri, astfel că totalul de locuri este

$$n' = \sum_{j=1}^{k} n_j$$

Dacă $n' > n$ atunci se adaugă $n' - n$ locuri imaginare. Dacă în împerecherea finală un student este asociat unui loc de admitere imaginar, această situatie va fi tratată ca si cum studentul a fost respins de oricare altă specializare astfel că se presupune că studentii de acest tip "optează" pentru această "specializare" ca ultimă optiune.

Alternativ, dacă $n < n'$, atunci se adaugă $n - n'$ studenti imaginari. Dacă un student imaginar este atribuit în final unei anumite specializări, aceasta corespunde admiterii de mai putini studenti decât câte locuri sunt disponibile la specializarea respectivă; se admite aici că fiecare specializare clasifică studentii imaginari ca ultimii.

Odată operate aceste modificări, vor fi exact atâtea locuri câti studenti sunt, asa încât de aici înainte se presupune că $n = \sum_{i=1}^{k} n_i$.

Apoi, se va descrie o instantă asemănătoare a problemei căsătoriilor. Se va considera fiecare loc ca un băiat si fiecare student ca o fată (rolul genurilor poate fi, desigur, inversat; în amble cazuri avem de-a face cu o împerechere stabilă). Locurile de admitere la aceeasi specializare au toate aceeasi listă de

1 La adresa http://en.wikipedia.org/wiki/Stable_marriage_problem se poate găsi formularea problemei. Tot acolo sunt date unele detalii ale algoritmului.

preferinte pentru studenti. Totodată, lista de preferinte a fiecărui student este extinsă astfel ca dacă "ea" preferă specializarea *A*, specializării *B*, atunci toate locurile de admitere la *A* sunt listate înaintea tuturor locurilor de admitere la *B* (De pildă, dacă *A* are 4 locuri si *B* are 2, atunci o parte a listei "ei" va fi … > A_1 > A_2 > A_3 > A_4 > … > B_1 > B_2 > …). În acest mod, se obtin *n băieti* si *n fete*, fiecare cu o listă de preferinte completă.
În final, se execută algoritmul TMA (Traditional Matching Algorithm) pentru a obtine potrivirea între studenti si locuri. Deoarece TMA produce totdeauna potriviri stabile, acesta produce admiteri stabile, adică nu există "mezaliante".

2. **Tăierea tortului**. Se consideră următorul protocol de împărtire a unui tort la trei persoane:
    1. A taie tortul în trei bucăti egale (egale conform aprecierii lui A)
    2. B taie fiecare din aceste trei bucăti în jumătate (cele două jumătati ale fiecărei treimi, egale după părerea lui B)
    3. Din aceste 6 bucăti, C alege cele mai bune două bucăti (după părerea lui)
    4. Din cele 4 bucăti rămase A alege două (cele mai bune după opinia lui)
    5. B ia ultimele două bucăti.

    (a) Este acest protocol echitabil pentru A? pentru B? pentru C?

Solutie: Protocolul este acceptabil pentru A si C, dar nu pentru B.
Este acceptabil pentru A deoarece el vine la alegere când rămân patru bucăti. Asta înseamnă că două din bucătile rămase trebuie să provină dintr-una din bucătile initiale pe care la-a tăiat chiar A, astfel că ele fac 1/3 din tort (presupunând că A a urmat corect protocolul).
Este acceptabil si pentru C deoarece C alege primul si poate alege cele mai mari două bucăti care pot constitui cel putin o treime din tort.
Nu este acceptabil pentru B. Se presupune că B atribuie celor trei bucăti tăiate initial de A valorile 1, 0 si 0. Apoi B taie bucata cea mare în două bucăti egale cu valoarea 1/2 fiecare. Dar dacă C ia acele două bucăti, B nu mai ia nimic din tort.

(b) Este acest protocol liber de suspiciune pentru A? pentru B? pentru C?

Solutie: Algoritmul este liber de orice suspiciune numai pentru C.
Deoarece C alege primul, el poate alege cele mai mari două bucăti, care totalizează cel putin cât orice altă pereche de bucăti din cele rămase. Astfel C nu poate suspecta că cineva l-a păcălit.
Algoritmul nu este lipsit de suspiciuni pentru B deoarece pentru el nu este acceptabil (după cum s-a arătat mai sus el se poate alege cu nimic, după măsura sa).
Algoritmul nu este lipsit de suspiciuni nici pentru A. Se admite că B taie fiecare din bucăti astfel încât A evaluează o bucată la 1/3 si pe cealaltă la 0. Acum se presupune că C ia două bucăti evaluate la 1/3 si pentru A rămâne numai 1/3.

1. Baza –2

Reprezentarea în baza –2 este analogă reprezentării în baza 2 (binară). Un întreg n este reprezentat ca $d_k d_{k-1} \ldots d_1 d_0$ dacă $n = d_k(-2)^k + d_{k-1}(-2)^{k-1} + \ldots + d_1(-2)^1 + d_0(-2)^0$ cu coeficientii d_j fie 0, fie 1.

a) Care este reprezentarea în baza –2 a numărului 9?

Solutie: 11001, deoarece $9 = 16 - 8 + 1$.

b) Demonstrati că orice întreg $n > 0$ poate fi reprezentat în baza –2.

Solutie: Se poate demonstra o proprietate mult mai tare si anume "orice întreg pozitiv sau negativ poate fi reprezentat în baza –2".

Fie $P(k)$ afirmatia: "Întregii de cel mult k cifre, care pot fi reprezentati în baza (negativă) –2, sunt cei 2^k întregi consecutivi care încep cu $(-2)^1 + (-2)^3 + \ldots + (-2)^{k-2}$ pentru k impar si $(-2)^1 + (-2)^3 + \ldots + (-2)^{k-1}$ pentru k par".

Adevărul acestei afirmatii se dovedeste prin inductie. Pentru cazul de bază $k = 1$, există $2 = 2^1$ întregi reprezentabili în baza –2 printr-un digit: 0 si 1. Pentru pasul inductiv $P(k) \Rightarrow P(k+1)$, se împarte demonstratia în două cazuri:

Cazul **1**. Dacă avem un k par, atunci putem reprezenta intregii de la p la $p + 2^k - 1$ prin cel mult k digiti, unde $p = (-2)^1 + (-2)^3 + \ldots + (-2)^{k-1}$. Pentru a avea întregii reprezentabili prin $k + 1$ digiti putem alege d_k egal cu 0 sau 1 pentru fiecare întreg din cei reprezentabili prin k digiti. Deoarece k este par, $(-2)^k = 2^k$. Dacă punem $d_k = 1$, atunci întregii de la $p + 2^k$ la $(p + 2^k - 1) + 2^k = p + 2^{k+1} - 1$ sunt reprezentabili prin $k + 1$ digiti. Dacă punem $d_k = 0$, atunci întregii de la p la $p + 2^k - 1$ sunt reprezentabili prin $k + 1$ digiti (acestia sunt întregii care se pot reprezenta prin k digiti). Astfel, întregii reprezentabili prin până la k + 1 digiti se întind de la p la $p + 2^{k+1} - 1$ care alcătuiesc o multime de 2^{k+1} întregi consecutivi.

Cazul **2**. Dacă numărul k este impar, putem reprezenta întregii de la p la $p + 2^k - 1$ prin cel mult k digiti, unde $p = (-2)^1 + (-2)^3 + \ldots + (-2)^{k-2}$. Pentru a avea întregii reprezentabili prin $k + 1$ digiti putem alege d_k egal cu 0 sau 1 pentru fiecare întreg din cei reprezentabili prin k digiti. Deoarece k este impar, $(-2)^k = -2^k$. Dacă punem $d_k = 1$, atunci întregii de la $p - 2^k$ la $(p + 2^k - 1) - 2^k = p - 1$ sunt reprezentabili prin $k + 1$ digiti, în plus fată de cei reprezentabili prin k digiti (care corespund punerii lui $d_k = 0$). Astfel, întregii reprezentabili prin până la $k + 1$ digiti se întind de la $p - 2^k$ până la $p + 2^k - 1$ si alcătuiesc o multime de 2^{k+1} întregi consecutivi. Începutul acestei secvente este

$$p - 2^k = (-2)^1 + (-2)^3 + \ldots + (-2)^{k-2} + (-2)^k$$

cum s-a presupus a fi în $P(k + 1)$.

Acum, că am dovedit că $P(k)$ se mentine pentru orice k, să arătăm că orice întreg pozitiv n poate fi exprimat în baza –2. Se găseste un k par pentru care $n \leq 2^k$. Multimea întregilor reprezentabili prin $k + 1$ digiti include atât pe 0 cât si pe

2^k. Deoarece am dovedit că această multime de întregi reprezentabili este consecutivă, n trebuie să fie reprezentabil în baza –2 prin cel mult $k + 1$ digiti.

c) Este reprezentarea în baza –2 unică? Cu alte cuvinte, poate orice întreg pozitiv n să fie reprezentat în exact o exprimare în baza –2? Demonstrati acest fapt si dati exemple.

Solutie: Răspunsul este afirmativ (cu conditia să nu considerăm diferite de pildă reprezentările 11001 si 011001 pentru numărul 9; ca de obicei, zerourile de la începutul unui număr sunt ignorate).

Să admitem că un acelasi întreg pozitiv n poate fi reprezentat în două moduri, fiecare utilizând cel mult k digiti. Acum luăm în considerare multimea întregilor reprezentabili cu până la k digiti. Am arătat mai devreme că această multime are 2^k întregi diferiti. Dacă vreunul din acesti întregi poate fi reprezentat în cel putin două moduri, acesti 2^k întregi au cel putin $2^k + 1$ reprezentări. Totusi, există cel mult 2^k reprezentăti posibile cu k digiti. Rezultă o contradictie, asa încât fiecare întreg pozitiv este reprezentat în mod unic în baza –2.

2. Notatia *O* (big-*O*)

Scopul acestei probleme este a explica notatia O într-un mod mai detaliat. Mai întâi se cuvin studiate următoarele:

Formal: Dacă $f(n)$ si $g(n)$ sunt două functii nenegative de variabila întreagă n, afirmatia $f(n) \in O(g(n))$ are semnificatia

$$\exists N_0 \in N.\ \exists C \in N.\ \forall x \in N.\ x \geq N_0 \Rightarrow 0 \leq f(x) \leq C.g(x)$$

Cu alte cuvinte, $O(g(n))$ este o multime de functii $\{f_i(n)$: $\exists N_0 \in N.\ \exists C \in N.\ \forall x \in N.\ x \geq N_0 \Rightarrow f_i(x) \leq C.g(x)\}$. Aceasta este definitia notatiei big-O.

Informal: Grosier vorbind, $f(n) \in O(g(n))$ are semnificatia "$f(n)$ nu creste cînd n creste, mai repede decât $g(n)$ (abstractie făcând de orice factor constant). De exemplu, $n^2 \in O(n^2)$, $n(n + 1)/2 \in O(n^2)$ si $10000n^2 \in O(n^2)$ deoarece toate aceste functii cresc asimptotic cu n, cu aceeasi rapiditate (dacă se ignoră factorii constanti). La fel, $n^2 \in O(n^3)$ deoarece n^2 creste mai lent decât n^3 pe măsură ce n creste.

Câteva relatii fundamentale:

Dacă $f(n) \in O(g(n))$ si $f'(n) \in O(g'(n))$ atunci $f(n) + f'(n) \in O(g(n) + g'(n))$.

Dacă $f(n) \in O(g(n))$ si $f'(n) \in O(g'(n))$ atunci $f(n) \times f'(n) \in O(g(n) \times g'(n))$.

Notatii obisnuite: Scrierea $f(n) \in O(g(n))$ este înlocuită adesea de scrierea $f(n) = O(g(n))$. Este un abuz de limbaj larg răspândit si frecvent întâlnit chiar în lucrări cu pretentii. De asemenea se scrie n^2 ca prescurtare pentru functia $f(n) = n^2$, asta mai ales pentru facilitarea scrierii.

Acum problema propriu-zisă:

(a) Demonstrati că $n^2 + 2007 \in O(n^3)$.

Solutie: Constantele $C = 1$ si $N_0 = 15$ satisfac definitia lui $O(\bullet)$ deoarece $0 \leq n^2 + 2007 \leq n^3$ este un adevăr pentru $n \geq 15$. Faptul poate fi verificat prin calcul. Derivata functiei de variabilă reală $u(x) = x^3 – x^2 – 2007$ este $u'(x) = 3n^2 – 2n$ si se poate verifica usor că $u(15) \geq 0$ si $u'(x) \geq 0$ pentru $x \geq 15$.

(b) Demonstrati că $100n^2 \log n \in O(n^3)$.

Solutie: Constantele $C = 100$ si $N_0 = 15$ satisfac definitia lui $O(\bullet)$. Din nou se recurge la verificarea prin calcul. Derivata functiei de variabilă reală $u(x) = 100x^3 - 100x^2\log x$ este $u'(x) = 300x^2 - 200x\log x - 100x$, $u''(x) = 600x - 200\log x - 300$ si $u'''(x) = 600 - 200/x$. Se poate verifica usor că $u(15) \geq 0$, $u'(15) \geq 0$, $u''(15) \geq 0$ si $u'''(x) \geq 0$ pentru $x \geq 15$. De aici rezultă că $u(n) \geq 0$ pentru orice $n \geq 15$.

(c) Adevărat sau fals? Există e natural astfel încât $2^n \in O(n^e)$)? Justificati pe scurt răspunsul.

Solutie: Fals.

O cale de a demonstra acest răspuns constă în evalurea limitei următoare recurgând de câte ori e necesar la regula lui l'Hopital:

$$\lim_{n \to \infty} \frac{n^e}{2^n} = \lim_{n \to \infty} \frac{en^{e-1}}{2^n \ln 2} = \cdots = \lim_{n \to \infty} \frac{e!}{2^n (\ln 2)^e} = 0$$

Asadar, n^e creste cu n asimptotic mult mai lent decât 2^n, ceea ce implică falsitatea afirmatiei din enunt.

În general, dacă $\lim_{n \to \infty} \frac{f(n)}{g(n)}$ există si este finită atunci $f(n) \in O(g(n))$. Dacă $\lim_{n \to \infty} \frac{f(n)}{g(n)}$ este infinită atunci $f(n) \notin O(g(n))$. Dacă aceeasi limită nu există nu se poate afirma nimic sigur.

(d) Demonstrati că dacă $f(n) \in O(g(n))$ si $g(n) \in O(h(n))$ atunci $f(n) \in O(h(n))$.

Solutie: Deoarece $f(n) \in O(g(n))$, există C_1 si N_1 pentru care $0 \leq f(n) \leq C_1.g(n)$ pentru orice $n \geq N_1$; deoarece $g(n) \in O(h(n))$, există C_2 si N_2 pentru care $0 \leq g(n) \leq C_2.h(n)$ pentru orice $n \geq N_2$. Prin combinarea acestor două inegalităti se obtine

$$0 \leq f(n) \leq C_1.C_2.h(n)$$

pentru orice $n \geq \max(N_1, N_2)$. Se poate lua atunci $C = C_1.C_2$ si $N_0 = \max(N_1, N_2)$ pentru a sustine că $f(n) \in O(h(n))$.

(e) Examinati critic argumentatia care urmează. Este rationamentul valid? Dacă nu, de ce nu? Dacă există o eroare, identificati pasul eronat si explicati ce este în neregulă.

Avem $n^2 = O(n^4)$.
Avem, de asemenea, $n^2 = O(n^3)$.
Prin tranzitivitate rezultă că $O(n^4) = O(n^3)$.
Aceasta înseamnă că $n^4 = O(n^3)$.

Solutie: Rescriem argumentatia în notatii adecvate. Se obtine:

Avem $n^2 \in O(n^4)$.
Avem, de asemenea, $n^2 \in O(n^3)$.
Prin tranzitivitate rezultă că $O(n^4) \in O(n^3)$.
Aceasta înseamnă că $n^4 \in O(n^3)$.

În pasul care se referă la tranzitivitate, justificarea este că $x \in A \wedge x \in B \Rightarrow A = B$. Acesta este un neadevăr. Contraexemplu: fie $A = N$, $B = Z$ (multimea numerelor naturale si multimea numerelor întregi) si $x = 0$. x apartine ambelor multimi, dar asta nu face multimile egale.

3. Detectarea unei puteri

(a) Proiectati un algoritm eficient pentru o functie numită *ispower?*, care, fiind dat un întreg pozitiv n si un altul $k < n$, testează dacă n este o putere a k perfectă. Asadar, *ispower?*(n, k) returnează *true* dacă si numai dacă $\exists\, x \in N$. $x^k = n$. Algoritmul propus ar trebui să ruleze în cel mai rău caz într-un timp $O((\log n)^c)$ cu c o constantă oarecare.

Solutie: Folosim o căutare binară pentru a afla dacă există pentru n o rădăcină de ordinul k.

ispower?(n, k):
initialize min = 1, max = n
while min ≠ max
set $i = \lfloor (\min + \max)/2 \rfloor$
if $i^k = n$ then return *true*
else if $i^k < n$ set max = $i - 1$
else (when $i^k > n$) set min = $i + 1$.
Return *false*.

(b) Verificati limita timpului de executie pentru algoritmul scris la punctul anterior.

După fiecare iteratie a buclei *while*, intervalul în care rădăcina de ordinul k s-ar putea situa este înjumătătit. Astfel, există cel mult $\log n$ iteratii ale buclei *while*. Pentru a calcula i^k, se poate folosi ridicarea la pătrat repetată care implică $O(\log k)$ multiplicări. Fiecare multiplicare ar putea consuma un timp de ordinul $O((\log n)^2)$. Timpul total de executie este $O(\log k(\log n)^3) = O((\log n)^4)$, deoarece $k < n$.

4. Cel mai mare divizor comun în binar

(a) Demonstrati că afirmatiile următoare sunt adevărate pentru orice $m, n \in N$.

i. Dacă m este par si n este par, atunci $\gcd(m, n) = 2\gcd(m/2, n/2)$.
Solutie: Dacă m si n sunt pare, atunci $\gcd(m, n) = 2\gcd(m/2, n/2)$ se demonstrează astfel: dacă d divide atât pe $m/2$ cât si pe $n/2$, atunci $2d$ divide pe m si n, astfel încât $2\gcd(m/2. n/2) | \gcd(m, n)$. În plus, $d' = \gcd(m, n)$ este par si deoarece d' divide pe m si pe n, $d'/2$ divide pe $m/2$ si pe $n/2$, de unde $d'/2 | \gcd(m/2, n/2)$ adică $\gcd(m, n) | 2\gcd(m/2, n/2)$.

ii. Dacă m este par si n este impar, atunci $\gcd(m, n) = \gcd(m/2, n)$.
Solutie: Faptul poate fi verificat ca mai sus: dacă d divide pe m si pe n atunci d este impar si astfel d divide pe $m/2$, de unde $\gcd(m, n) | \gcd(m/2, n)$. Implicatia inversă este evidentă. Cazul cu m impar si n par este simetric.

iii. Dacă m si n sunt ambele impare si $m \geq n$, atunci $\gcd(m, n) = \gcd((m - n)/2, n)$.
Solutie: Se stie că dacă $m \geq n$, $\gcd(m, n) = \gcd(m - n, n)$ – aceasta este baza algortimului lui Eulid. Deoarece $m - n$ este par si n este impar,

prin aplicarea rezultatului de la punctul anterior se obtine gcd(m, n) = gcd($m - n$, n) = gcd(($m - n$)/2, n).

(b) Scrieti un algoritm care să calculeze gcd(m, n) care să folosească cel mult $O(\log m + \log n)$ scăderi, înjumătătiri, dublări si verificări de paritate.

Solutie: Un algoritm posibil:

gcd(m, n):

if $m = 0$, returnează n. if $n = 0$, returnează m.

if m este par si n este par, returnează 2gcd(m/2, n/2).

if m este par si n este impar, returnează gcd(m/2, n).

if m este impar si n este par, returnează gcd(m, n/2).

if $m \leq n$ then returnează gcd(($n - m$)/2, m) else returnează gcd(($m - n$)/2, n).

Acest a algoritm foloseste cel mult $O(\log m + \log n)$ scăderi, înjumătătiri, dublări si verificări de paritate.

Demonstratie: Mai întâi mentinem invariantul că la fiecare apel recursiv parametrii sunt ambii nenegativi (singurul caz netrivial este cel din pasul ultim, dar acolo din faptul că numerele m si n sunt impare decurge că $|m - n|$ este divizibil cu 2 si deci $|m - n|/2$ este un întreg nenegativ).

În al doilea rând observăm că fiecare apel rcursiv reduce unul din argumente printr-un factor de doi sau mai mare. Astfel, algoritmul trebuie să se sfârsească când unul din argumente se anulează (nu e posibilă o scădere eternă). În fapt, cel mult $\log m + \log n$ iteratii recursive sunt suficiente.

În al treilea rând fiecare din aceste iteratii face cel mult 6 verificări de paritate, 2 diviziuni prin 2, o multiplicare cu 2 si o scădere, adică un număr constant din aceste operatii.

Deoarece sunt $O(\log m + \log n)$ iteratii, fiecare cu executarea unui număr constant din operatiile de mai sus, numărul total de astfel de operatii executate este $O(\log m + \log n)$.

De observat că sunt motive multiple pentru care ar putea fi preferat algoritmul gcd binar si nu algoritmul lui Euclid.

Un motiv posibil: s-a arătat că algoritmul gcd binar rulează într-un timp pătratic (el execută un număr liniar de operatii, fiecare consumând un timp liniar), iar algoritmul lui Euclid consumă, se pare, un timp ceva mai mare. Chiar dacă analiza algoritmului lui Euclid este întrucâtva marcată de pesimism, cu un pic de efort se poate aduce si acesta la un timp pătratic. Diferenta dintre cele două căi de stabilire a gcd constă mai curând în simplitatea analizei timpului în cazul gcd binar.

În practică, un motiv mai important este acela că factorii constanti din algoritmul binar par a fi mai mici si operatiile utilizate în algoritmul binar sunt mai potrivite implementării pe calculator. Divizarea prin 2, multiplicarea cu 2 si testarea par/impar sunt opertii extrem de rapide când operanzii sunt reprezentati în binar si asta face algoritmul binar mult mai atractiv din punct de vedere software. Algoritmul este atractiv potential si din punct de vedere hardware deoarece nu trebuie implementat un circuit de reducere modulară.

5. Ecuatii diofantice

Fiind dati întregii pozitivi a, b, c să se găsească o solutie întreagă a ecuatiei $ax + by + cz = 1$ (necunoscutele sunt x, y, z).

(a) Proiectati un algoritm eficient pentru a găsi o asemenea solutie, în ipoteza $\gcd(a, b) = \gcd(b, c) = \gcd(c, a) = 1$.

Solutie: Se pune $z = 0$. Apoi se utilizează algoritmul lui Euclid extins pentru a găsi întregii x, y care satisfac relatia $ax + by = 1$.

(b) Proiectati un algoritm eficient pentru a găsi o asemenea solutie, în ipoteza mai compactă $\gcd(a, b, c) = 1$.

Solutie: Fie $d = \gcd(a, b)$. Se execută algoritmul lui Euclid extins pentru a stabili întregii x, y care satisfac relatia $ax + by = d$. Se observă că $\gcd(c, d) = 1$ deoarece $\gcd(a, b, c) = 1$. Apoi se poate aplica din nou algortimul lui Euclid extins pentru a găsi întregii z, w pentru care $cz + dw = 1$. Cu rezultatul deja obtinut, $d = ax + by$, avem $cz + dw = cz + (ax + by)w = awx + bwy + cz = 1$. Solutia este (wx, wy, z).

1. Se presupune că cineva alege o valoare pentru n care nu este un produs de două numere prime, adică $n = pq$ cu $p > 1$, $q > 1$ si q este compus. Ar fi, evident, mai usor de factorizat ceea ce ar pune o problemă de risc sub aspectul securitătii. Dar vor functiona operatiile de criptare si de decriptare cu acest n? Sustineti răspunsul.

Solutie: Nu, criptarea/decriptarea nu mai functionează. Pentru a avea un contraexemplu, fie $n = 45 = 5.9$. Dacă punem $p = 5$ si $q = 9$, atunci pentru o cheie de criptare $(e, 45)$, am alege o cheie de decriptare $(d, 45)$ pentru care $de \equiv 1 \pmod{(5-1)(9-1)}$, adică

$$de \equiv 1 \pmod{32}$$

Astfel, dacă $e = 5$, inversul mod 32 este $d = 13$. Dar dacă criptăm si decriptăm mesajul $M = 2$, obtinem

$$(2^5)^{13} = 2^{65} \equiv 32 \pmod{45}.$$

Mai general, RSA se bazează pe faptul că $(M^e)^d \equiv 1 \pmod{n}$, deoarece această relatie este cea care asigură că decriptarea este inversa criptării. Relatia se va mentine dacă $de \equiv 1 \pmod{\varphi(n)}$. Dar, dacă q este compus, $\varphi(n) \neq (p-1)(q-1)$ si astfel vom avea propozitia $de \equiv 1 \pmod{(p-1)(q-1)}$ falsă. În consecintă, în cazul exemplificat criptarea RSA nu poate lucra.

2. Se consideră rezultatul următor, demonstrat prima oară cu multe secole în urmă:

Teorema 1 (**Euclid**): *Există o infinitate de numere prime*.

Demonstratie: Se admite contrarul adică numerele prime sunt în număr finit. Fie acestea, în ordine crescătoare, $p_1 = 2$, $p_2 = 3$, $p_3 = 5$, …, p_k. Fie $q_k = p_1p_2p_3\ldots p_k + 1$. Observăm că q_k este un număr nou care nu este în lista de numere prime p_1, …, p_k. În acelasi timp el nu este divizibil cu vreunul din numerele p_i deoarece $q_k \equiv p_1p_2p_3\ldots p_k + 1 \equiv 1 \pmod{p_i}$, ceea ce înseamnă că q_k este un număr prim nou diferit de p_1, …, p_k. Apare contradictia care încheie demonstratia.

□

Fie p_1, …, p_k primele k numere prime.

(a) Suntem siguri că $p_1p_2p_3\ldots p_k + 1$ este totdeauna prim pentru orice $k \geq 1$?

Solutie: Nu. Se consideră cazul în care se multiplică primele 6 numere prime si se adaugă 1. Se obtine

$$2.3.5.7.11.13 + 1 = 30031 = 59.509$$

ceea ce pare a fi un contraexemplu convingător.

(b) Este demonstratia teoremei de mai sus validă? Explicati.

Solutie: Demonstratia este încă validă în ciuda faptului că numărul $q_k = p_1p_2p_3\ldots p_k + 1$ nu este cu necesitate prim. Fie P_k propozitia conform căreia lista completă a numerelor prime este p_1, …, p_k si Q_k propozitia după care q_k este prim. Demonstratia a arătat că implicatia $P_k \Rightarrow Q_k$ este adevărată. Demonstratia

acestei implicatii este validă si implicatia este adevărată pentru orice k. La punctul (a) am descoperit că Q_6 este falsă. Cu toate că acest fapt ne poate contraria, nu este vorba de o contradictie: rezolvarea paradoxului este aceea că P_6 si Q_6 sunt ambele false astfel că $P_6 \Rightarrow Q_6$ este adevărată (deoarece *false* implică orice).

Să încercăm o altă cale. Stim că propozitia $P_k \Rightarrow Q_k$ este echivalentă logic cu $\neg P_k \vee Q_k$. Astfel, iată un alt mod de a judeca demonstratia. Demonstratia arată că pentru orice k fie (a) $p_1, \ldots, p_k$ nu este lista completă a numerelor prime, fie (b) q_k este un număr prim nou, diferit de cele din lista $p_1, \ldots, p_k$. În oricare din cele două cazuri $p_1, \ldots, p_k$ nu este lista completă a numerelor prime. Deoarece acesta este un adevăr pentru orice k, asta înseamnă că nici o listă finită nu poate cuprinde toate numerele prime, sau echivalent, există un număr infinit de numere prime.

3. Se avansează spre rezolvare următorele:

(a) Se consideră amprenta $F_p(w) = w \bmod p$, prin care întregul $w = 2^{n-1}w_{n-1} + \ldots + 2w_1 + w_0$ se identifică cu sirul de n biti $w = (w_0, w_1, \ldots, w_{n-1})$.

Fie $n, m \in N$ cu $n < m$. Fie X un sir de m biti $(x_0, x_1, \ldots, x_{m-1})$. Fie $X_{(i)} = (x_i, x_{i+1}, \ldots, x_{i+n-1})$ un subsir de n biti al lui X, subsir care începe la pozitia i, sau echivalent, numărul întreg $X_{(i)} = 2^{n-1}x_{i+n-1} + \ldots + 2x_{i+1} + x_i$. De exemplu, dacă $m = 5$, $n = 4$ si $X = (0, 1, 1, 0, 1)$, atunci $X_{(0)} = (0, 1, 1, 0) = 6$ si $X_{(1)} = (1, 1, 0, 1) = 11$.

Arătati cum se calculează eficient $F_p(X_{(i)})$ din $F_p(X_{(i+1)})$.

Solutie: Se observă că

$$X_{(i)} = 2^{n-1}x_{i+n-1} + \ldots + 2x_{i+1} + x_i =$$
$$= 2(2^{n-1}x_{i+n} + 2^{n-2}x_{i+n-1} + \ldots + x_{i+1}) - 2^n x_{i+n} + x_i =$$
$$= 2X_{(i+1)} - 2^n x_{i+n} + x_i$$

Relatia aceasta se mentine si modulo p, astfel că se poate scrie

$$F_p(X_{(i)}) = 2F_p(X_{(i+1)}) - 2^n x_{i+n} + x_i \bmod p$$

Se poate pre-calcula $-2^n \bmod p$. Apoi calculul lui $F_p(X_{(i)})$ din $F_p(X_{(i+1)})$ necesită numai trei adunări si până la două scăderi (pentru reducerea la modulo p), ceea ce consumă un timp $O(\log p)$ indiferent de cât de mare este n.

(b) Se presupune că sunt date un sir X de m biti si un sir Y de n biti si se urmăreste testarea relatiei "Y este subsir al lui X". Se consideră următorul algoritm naiv:

Naivestringmatch(X, Y):

1. For $i = m - n$ down to 0 do:
2. If $X_{(i)} = Y$, return i.
3. Return "no match".

Argumentati că acest algoritm naiv are în cel mai rău caz un timp de executie $O(mn)$ dacă se numără fiecare operatie la nivel de bit ca o unitate de timp.

Solutie: Pasul al doilea compară două siruri de n biti, reclamă asadar $O(n)$ operatii. Bucla este executată de $m - n + 1$ ori, ceea ce înseamnă $O(m)$ (uzual,

m ar trebui să fie mult mai mare decât n). Asadar, timpul consumat de algoritm este de $O(mn)$ operatii cu biti.
Cazul cel mai rău se poate întâlni, desigur, în practică; de pildă când Y este alcătuit numai din zeoruri si X este alcătuit aproape numai din zerouri.
(c) Produceti un algoritm mai rapid (prin eficientizarea pasului al doilea).
Solutie:

Fasterstringmatch(X, Y):

0. Se alege la întâmplare un număr prim p. Se calculează $F_p(Y)$ si $F_p(X_{(m-n)})$.
1. For $i = m - n$ down to 0, do:
2. If $F_p(X_{(i)}) = F_p(Y)$ then do:
3. If $F_p(X_{(i)}) = F_p(Y)$ then do:
4. Return i.
5. Return "no match".

Fiecare iteratie din buclă calculează $F_p(X_{(i)})$ din valoarea precedentă $F_p(X_{(i+1)})$ pe baza celor arătate la punctul (a). Astfel, pasul 2 devine foarte rapid: $O(\log p)$. Mai mult, dacă $X_{(i)} \neq Y$, atunci probabil $F_p(X_{(i)}) \neq F_p(Y)$ si acest pas 3 costisitor va fi executat de un număr de ori mai mic. Desi nu se poate defini o limită pentru un cel-mai-rău-caz sub aspectul timpului de calcul, foarte probabil acest algoritm va fi practic mult mai rapid decât cel anterior dacă $m < p < 2^n$.
De observat că si cu acest algoritm modificat este teoretic posibil ca timpul de calcul să fie $O(mn)$ dar acest cel-mai-rău-caz depinde numai de alegerile interne aleatoare făcute de algoritm si nu de intrări. Asadar, nu există un cel-mai-rău-caz din punct de vedere al intrărilor. Pentru orice intrare acest algoritm va lucra mai repede decât $O(mn)$.
În fapt, dacă în pasul 0 se ia p un număr prim de $4\log m$ biti, atunci durata executiei este aproape totdeauna mărginită de $O(m\log m)$ cu exceptia unui caz nefericit, probabil în proportia de cel mult $1/m$ (a se vedea si punctul următor). Nu se cere demonstrarea acestui fapt.
(d) Se presupune că algoritmului i se permite sansa mică (să spunem, mai mică decât $1/m$) de a returna un răspuns gresit. Descrieti un algoritm cu un timp de rulare pentru cel-mai-rău-caz de $O(m\log m)$. Analiza poate fi întrucâtva informală dar asigurati-vă că arătati toate alegerile de parametri.
(Dacă nu puteti găsi un algoritm cu o asemenea limită demonstrabilă pentru timpul de calcul, reduceti timpul de rulare asimptotic cât de mult puteti. Ignorati factorii constanti.)

Solutie: Fie Fasteststringmatch aceeasi rutină ca Fasterstringmatch dată mai devreme cu exceptia omiterii pasului 3 si cu alegerea în pasul 0 a unui număr prim p aleator de $4\log m$ biti. Durata rulării este de cel mult $O(m\log m)$ deoarece fiecare iteratie a buclei se execută într-un timp $O(\log p) = O(4\log m) = O(\log m)$ (a se revedea punctul (a)) si sunt $O(m)$ iteratii.
Probabilitatea ca amprenta să fie înselătoare (adică pasul 3 s-ar executa dar nu pasul 4) este de cel mult $1/m^2$ pentru o parcurgere a buclei. Acesta este cazul unic în care Fasteststringmatch poate genera un răspuns eronat. Mai mult, sunt

numai $m - n + 1$ iteratii în buclă, astfel încât sansa totală a unei erori este de cel mult $(m - n + 1)/m^2 \leq 1/m$.

1. Algoritmul de calcul al puterii a^b mod c prin ridicare la pătrat repetată nu conduce în mod necesar la numărul minim de multiplicări. Dati un exemplu pentru b ($b > 10$) astfel încât calculul să fie executat în mai putine multiplicări, printr-o metodă diferită.

Solutie: Se consideră $b = 15$. Algoritmul prin repetarea ridicării la pătrat execută 6 multiplicări:

$$a.a = a^2;\ a.a^2 = a^3;\ a^3.a^3 = a^6;\ a^6.a = a^7;\ a^7.a^7 = a^{14};\ a^{14}.a = a^{15}$$

Totusi, se poate calcula a^{15} si în numai 5 multiplicări:

$$a.a = a^2;\ a^2.a^2 = a^4;\ a^4.a = a^5;\ a^5.a^5 = a^{10};\ a^{10}.a^5 = a^{15}$$

La fel, se pot utiliza ridicări la cub reptate în loc de ridicări la pătrat si se obtine o reducere de multiplicări pentru unii exponenti b. De pildă, pentru a evalua a^{27}, ajung sase multiplicări:

$$a^3 = a.a.a;\ a^9 = a^3.a^3.a^3;\ a^{27} = a^9.a^9.a^9$$

Comentariu: În comunitatea cercetătorilor s-a studiat o generalizare a problemei găsirii unor *lanturi aditive* scurte pentru ridicarea la putere modulară. Un lant aditiv este o secventă de exponenti începând cu 1 si terminându-se cu b, în care fiecare întreg din secventă poate fi obtinut ca suma a doi întregi din secventă precedenti. De pildă, calculul prin ridicare succesivă la pătrat pentru $b = 15$ corespunde lantului aditiv 1, 2, 3, 6, 7, 14, 15; calculul optimizat de pe linia următoare corespunde lantului aditiv 1, 2, 4, 5, 10, 15. Formal, secventa b_0, b_1, …, b_l este un lant aditiv dacă $b_0 = 1$ si pentru orice $k \in \{1, \ldots, l\}$ există $i, j \in \{0, \ldots, i - 1\}$ astfel încât $b_k = b_i + b_j$. Valoarea l este numită lungimea lantului aditiv. Orice lant aditiv de lungime l care se sfârseste cu b produce imediat o metodă de exponentiere pentru puterea a b prin l multiplicări.

S-a consumat un effort de cercetare apreciabil pentru obtinerea de algoritmi capabili a găsi lanturi aditive. Pentru a da un sens rezultatelor din domeniu: nici un lant aditiv pentru b nu poate fi mai scurt decât $\log b$; prin ridicarea la pătrat repetată se obtine o lungime de lant de cel mult $2\log b$ (si $1{,}5\log b$, în medie); există alti algoritmi care dau lanturi ceva mai scurte; dar este o problemă deschisă construirea de algoritmi în timp polinomial care, la intrarea b, să producă cele mai scurte lanturi aditive posibile pentru b.

2. Fie p si q numere prime si $N = pq$. Arătati cum se pot determina p si q fiind date N si $(p - 1)(q - 1)$. Cu alte cuvinte, fiind dată cheia publică (e, N), e exponentul de criptare si N modulul RSA, precum si valoarea $\varphi(N) = (p - 1)(q - 1)$, este posibil a calcula p si q prin operatii algebrice simple (în timp polinomial). Aceasta arată că determinarea lui $\varphi(N)$ este la fel de grea ca si factorizarea.

Solutie: Fiind date $pq = N$ si $(p - 1)(q - 1) = \varphi(N)$, urmează a se afla p si q. Se poate pune $q = N/p$ după care o înlocuire în ecuatia a doua produce după câteva prelucrări algebrice ecuatia în p

$$p^2 + (\varphi(N) - N - 1)p + N = 0$$

care poate fi rezolvată aplicând formula cunoscută. Dacă p a fost găsit, este usor a-l găsi si pe q.

3. Problemă cu cadre didactice si criptare

(a) Se presupune că la un anumit curs activează trei profesori si doi asistenti. Solutiile la tema de casă următoare sunt criptate printr-o cheie de criptare în uzul tuturor celor cinci. Cei trei profesori, sau un asistent si un profesor, sau ambii asistenti trebuie să poată accesa solutiile. Sugerati o schemă de împărtire a secretului care să asigure aceste cerinte (*Indicatie*: Încercati ponderi.).

Solutie: Utilizati o schemă partajată 3-din-7 pentru a genera 7 părti ale acestui secret. Dati o parte fiecărui profesor, si două părti fiecărui asistent. Cei trei profesori laolaltă au 3 părti; un profesor si un asistent au împreună 3 părti si cei doi asistenti au 4 părti; astfel, în orice caz, aceste subseturi au părti suficiente pentru a recupera secretul.

Mai precis, imaginati-vă secretul ca ordonata la origine a unui polinom de gradul al doilea modular $f(x)$. Fiecărui profesor i se dă câte un punct pe $f(x)$ si fiecărui asistent câte două puncte pe curba $f(x)$, punctele fiind diferite în perechi. Pentru a descifra $f(x)$ si ordonata ei la origine sunt necesare cel putin trei puncte. Acest minim de puncte poate fi cumulat numai de trei profesori, de un profesor si un asistent si de cei doi asistenti. Acelasi minim nu poate fi întrunit numai de doi profesori sau numai de un asistent.

(b) Se presupune acum că grupa de cursanti este instruită de trei profesori, fiecare cu doi asistenti proprii. Oricare doi profesori pot accesa datele atât timp cât unul din asistentii fiecărui profesor (un total de cel putin patru persoane) este de asemenea prezent. Acum cum se procedează?

Solutie: Utilizati un sistem de partajare a secretului 2-din-3 pentru a genera părtile s_1, s_2, s_3 ale secretului. Apoi, utilizati o schemă de partajare 4-din-5 pentru a genera sub-părtile $s_{1,1}$, …, $s_{1,5}$ ale lui s_1, tratând s_1 ca secretul (unic); se dau trei sub-părti primului profesor si o subparte fiecăruia din asistentii lui. Se procedează analog cu ceilalti profesori.

Iată o descriere mai exactă. Mai întâi să luăm la întâmplare un polinom de gradul 1, $P(x)$, astfel încât secretul este $P(0)$. Apoi luăm tot la întâmplare trei polinoame de gradul 3, $Q_1(x)$, $Q_2(x)$ si $Q_3(x)$, astfel încât $Q_i(0) = P(i)$ pentru oricare $i = 1, 2, 3$. Cu alte cuvinte, secretul fiecărui polinom $Q_i(x)$ este una din părtile derivate din partajarea secretului bazat pe $P(x)$.

Acum pentru fiecare $Q_i(x)$ generăm 5 sub-părti $Q_i(1)$, …, $Q_i(5)$, atribuind 3 sub-părti profesorului si câte una asistentilor lui. Atunci, pentru a recupera mesajul, fiecare profesor i se întruneste cu unul din asistentii săi si recuperează secretul polinomului $Q_i(x)$, care este partea lui din secretul polinomului $P(x)$. În final, doi profesori se pot întruni pentru a recupera $P(0)$.

4. O problemă de calcul modular

(a) Demonstrati următoarele: Dacă p este un număr prim si $y_1, \ldots, y_n \in N$ sunt numere diferite de 0 modulo p, atunci si $y_1 \times \ldots \times y_n$ este diferit de 0 modulo p.

Solutie: Deoarece $y_1, \ldots, y_n \in N$ sunt numere diferite de 0 modulo p, p nu este factor pentru nici unul din ele. Astfel, factorizarea la factori primi a numărului $y_1 \times \ldots \times y_n$ nu include pe p, asadar p nu divide acest număr si

$$y_1 \times \ldots \times y_n \neq 0 \pmod{p}$$

Solutie alternativă: Deoarece $y_1, \ldots, y_n \in N$ sunt toate numere diferite de 0 modulo p, ele au toate inverse modulo p. Fie $z = y_1^{-1} \times \ldots \times y_n^{-1} \bmod p$. Atunci $z \times (y_1 \times \ldots \times y_n) = 1 \neq 0 \pmod{p}$ ceea ce înseamnă că p nu divide pe $z \times (y_1 \times \ldots \times y_n)$ ceea ce la rându-i implică faptul că p nu divide pe $y_1 \times \ldots \times y_n$.

(b) Demonstrati că fiind dat un număr prim p si doi întregi a si b, este totdeauna posibil a găsi un polinom $f(x)$ de grad cel mult 1, astfel încât $f(0) \equiv a \pmod{p}$ si $f(1) \equiv b \pmod{p}$.

Solutie: Fie $f(x) = a + (b - a)x$. Se verifică usor că $f(0) = a \equiv a \pmod{p}$ si $f(1) = a + (b - a) = b \equiv b \pmod{p}$.

(c) Se dau un număr prim p si un număr pozitiv $n < p$. Arătati cum se stabileste un polinom $f(x)$ de gradul cel mult n care să satisfacă $f(0) \equiv f(1) \equiv \ldots \equiv f(n - 1) \equiv 0 \pmod{p}$. Cu alte cuvinte polinomul f trebuie să fie congruent cu zero în punctele $x = 0, \ldots, n - 1$; pentru $x = n$ polinomul trebuie să fie 1 mod p. *Indicatie*: Se consideră $F(x) = (x - 0)(x - 1)\ldots(x - (n - 1))$. Ce puteti spune despre el?

Solutie: Fie $F(x) = (x - 0)(x - 1)\ldots(x - (n - 1))$ si se defineste $a = F(n) \bmod p$. Se observă că $a \equiv n! \pmod{p}$ este inversabil modulo p deoarece $n < p$ (avem $a^{-1} \equiv n^{-1} \times (n - 1)^{-1} \times \ldots \times 1^{-1} \pmod{p}$ si fiecare din numerele $1, \ldots, n$ este inversabil modulo p deoarece sunt mai mici decât p). Fie $b = a^{-1} \pmod{p}$. Acum punem

$$f(x) = bF(x)$$

Vom avea $f(0) \equiv f(1) \equiv \ldots \equiv f(n - 1) \equiv 0 \pmod{p}$ deoarece $F(0) = F(1) = \ldots = F(n - 1) = 0$. Vom avea totodată $f(n) \equiv F(n)^{-1} F(n) \equiv 1 \pmod{p}$. $F(n)$ este de gradul n conform definitiei, asadar f are gradul cel mult n. Această alegere a lui f satisface toate cerintele.

(d) Se dau p si n ca mai sus dar se dă de asemenea un indice j cu $0 \leq j \leq n$. Stabiliti un polinom $g_j(x)$ de grad cel mult n care să satisfacă conditia

$$g_j(i) \equiv \begin{cases} 0 & \text{daca } i \neq j \\ 1 & \text{daca } i = j \end{cases} \pmod{p} \quad \text{pentru orice } i = 0, 1, \ldots, n$$

Cu alte cuvinte, polinomul g_j trebuie să fie congruent cu 0 în punctele $x = 0, \ldots, n$ cu exceptia punctului $x = j$ unde trebuie să fie congruent cu 1 (mod p).

Solutie: Se folseste aceeasi idee ca si la punctul anterior. Se defineste $G_j(x) = (x - 0)\ldots(x - (j - 1))(x - (j + 1))\ldots(x - n)$ si $g_j(j) = (G_j(j)^{-1} \bmod p) G_j(j)$. Ca si mai sus, $g_j(i) = 0$ pentru $i \neq j$, $g_j(j) \equiv 1 \pmod{p}$ si g_j are gradul cel mult n, asadar satisface conditiile cerute.

(e) Se dă un număr prim p, un număr n, $0 < n < p$ si o secventă de valori $a_0, a_1, \ldots, a_n \pmod{p}$. Descrieti un algoritm eficient de stabilire a unui polinom

$h(x)$ de gradul cel mult n care să satisfacă $h(0) \equiv a_0 \pmod p$, $h(1) \equiv a_1 \pmod p$, …, $h(n) \equiv a_n \pmod p$.

Indicatie: Ce puteti spune despre polinomul $3g_0(x) + 7g_1(x)$ cu $g_0(x)$, $g_1(x)$ definite la punctul (d)? Vreo idee?

Solutie: Fie

$$h(x) = a_0g_0(x) + a_1g_1(x) + \ldots + a_ng_n(x)$$

cu polinoamele g_j definite la punctul (d). Polinomul h are gradul cel mult n deoarece polinoamele g_j au gradul cel mult n si, în plus, $h(i) \equiv a_0g_0(i) + a_1g_1(i) + \ldots + a_ng_n(i) \equiv 0 + \ldots + a_i.1 + 0 + \ldots + 0 \equiv a_i \pmod p$ precum se doreste.

De observat că h poate fi calculat eficient. Multiplicarea unui polinom de gradul d cu $(x - i)$ modulo p necesită d multiplicări si d adunări modulo p, astfel că fiecare $G_j(x)$ se poate calcula într-un timp $O(n^2(\log p)^2)$. Inversarea lui $G_j(j)$ modulo p poate fi făcută într-un timp $O((\log p)^3)$. Asadar, toti g_j se pot calcula în $O(n^3(\log p)^2 + n(\log p)^3)$, ceea ce dă un timp total de calcul al lui h modulo p de ordinul $O(n^3(\log p)^2 + n^2(\log p)^2 + n(\log p)^3)$. Deoarece intrarea este lungă de $n\log p$ biti, aceasta arată că algoritmul are un timp de rulare polinomial în lungimea intrării (cel mult cubic) ceea ce face ca algoritmul să poată fi considerat eficient.

5. Numărătoare

Câte dintre sirurile de biti de lungime 10 contin fie (a) cel puti cinci de 0 consecutivi sau (b) cel putin cinci de 1 consecutivi? Detaliati solutia.

Solutie: Se calculează mai întâi secventele cu cel putin cinci de 0 consecutivi. O strategie de numărare se bazează pe locul unde secventa începe. Ea poate începe oriunde între primul digit si al saselea, inclusiv. Dacă secventa începe la primul digit, primii cinci digiti sunt zero si sunt $2^5 = 32$ de alegeri pentru ceilalti 5 digiti. Dacă secventa începe după primul digit, ea trebuie precedată de un 1. Ceilalti patru digiti pot fi alesi liber în $2^4 = 16$ moduri. Astfel numărul total de siruri de 10 biti cu cel putin cinci zerouri consecutive este $2^5 + 5.2^4 = 112$.

Prin simetrie, există 112 de siruri de 10 biti cu cel putin cinci unităti binare consecutive.

Două siruri contin ambele secvente simultan: 0000011111 si 1111100000. Numărul de siruri care cumulează cele două proprietăti (a) si (b) este atunci $112 + 112 – 2 = 222$.

1. Coeficienti binomiali

a. Fie S o multime cu 10 elemente. Câte submultimi distincte de câte 4 elemente are S?

Răspuns: $C_{10}^4 = 210$ submultimi

b. Coeficientii binomiali satisfac multe identităti utile. Una dintre ele este $C_n^m = C_n^{n-m}$. Alta este $C_n^m + C_n^{m+1} = C_{n+1}^{m+1}$. Demonstrati cele două identităti pe cale algebrică.

Solutie: Pentru prima identitate se obtine succesiv

$$C_n^{n-m} = \frac{n!}{(n-m)!(n-(n-m))!} = \frac{n!}{(n-m)!m!} = C_n^m$$

Pentru a doua identitate

$$C_n^m + C_n^{m+1} = \frac{n!}{m!(n-m)!} + \frac{n!}{(m+1)!(n-m-1)!} =$$

$$= \frac{n!}{m!(n-m-1)!}\left(\frac{1}{n-m} + \frac{1}{m+1}\right) = \frac{(n+1)!}{(m+1)!(n-m)!} = C_{n+1}^{m+1}$$

c. Demonstrati că $C_n^0 + C_n^1 + ... + C_n^n = 2^n$

Solutie: $2^n = (1+1)^n = \sum_{j=0}^{n} C_n^j 1^{n-j} 1^j = \sum_{j=0}^{n} C_n^j$

d. O altă identitate importantă este $C_m^m + C_{m+1}^m + C_{m+2}^m + ... + C_n^m = C_{n+1}^{m+1}$. Demonstrati-o.

Solutie combinatorială: Fie o grupă de $n + 1$ studenti, numerotati chiar asa, de la 1 la $n + 1$. Numărul de subgrupe de câte $m + 1$ studenti este C_{n+1}^{m+1}. Aceste subgrupe se pot enumera si prin nominalizarea studentului $j + 1$ ca lider de subgrupă si apoi prin alegerea celorlalti m studenti dintre cei j cu numere mai mici. Numărul total de subgrupe este

$$\sum_{j=m}^{n} C_j^m = C_{n+1}^{m+1}$$

Solutie algebrică: Se face o demonstratie prin inductie după n. Identitatea este adevărată pentru $n = m$: $C_m^m = C_{m+1}^{m+1}$. Pentru pasul inductiv se consideră identitatea adevărată pentru $n = k$. Se demonstrează că este adevărată si pentru $n = k + 1$.

$$\sum_{j=m}^{k+1} C_j^m = C_{k+1}^m + \sum_{j=m}^{k} C_j^m = C_{k+1}^m + C_{k+1}^{m+1} = C_{k+2}^{m+1}$$

prin efectul ipotezei inductive si prin efectul punctului b. de mai sus

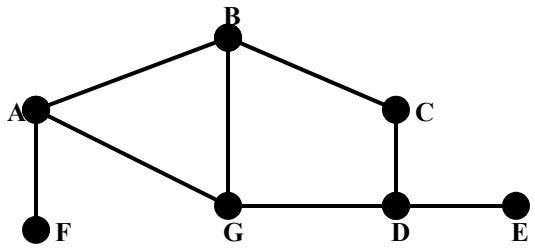
e. Fie $S_n = 1.2.3 + 2.3.4 + \ldots + (n-2)(n-1)n$. Stabiliti o expresie condensată pentru S_n.

Solutie:

$$S_n = \sum_{j=3}^{n} \frac{j!}{(j-3)!} = \sum_{j=3}^{n} 3!C_j^3 = 3!C_{n+1}^4 = \frac{(n-2)(n-1)n(n+1)}{4}$$

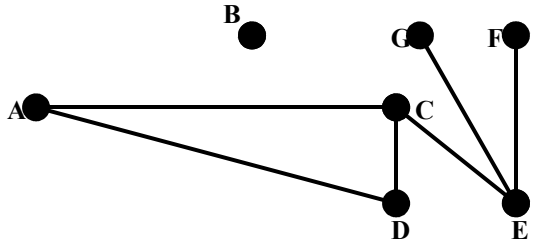
uzând de rezultatul de la punctul anterior.

2. Se consideră graful din figură. Scrieti matricea de adiacentă. Ordonati nodurile alfabetic.



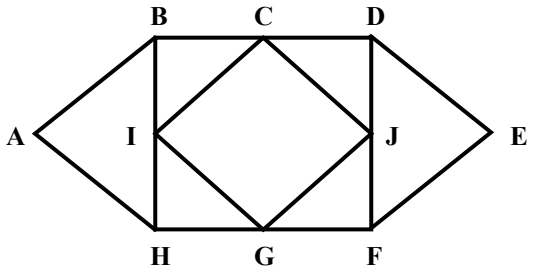
3. Sapte studenti merg în vacantă. Ei decid ca fiecare să trimită câte o ilustrată altor trei din ceilalti. Este posibil ca fiecare student să primescă ilustrate de la exact cei trei cărora le-a trimis ilustrate? Sustineti afirmatia.

4. Se consideră graful din figură. Scrieti matricea de adiacentă. Ordonati nodurile alfabetic.



5. Determinati dacă există un graf simplu cu opt noduri care să aibă secventa de grade 6, 5, 4, 3, 2, 2, 2, 2.

6. Determinati dacă graful alăturat este eulerian. Dacă este asa stabiliti un drum eulerian, dacă nu explicati de ce.



Formula lui Euler. Pentru orice reprezentare a unui graf plan conex

$$v - e + f = 2$$

Unde v este numărul de noduri, e este numărul de muchii si f este numărul de fețe.

Definiție: Un *drum eulerian* într-un graf este un drum care utilizează fiecare arc exact o dată. Dacă un asemenea drum există, se spune că graful este *traversabil*.
Definiție: Un *ciclu eulerian* este un ciclu care utilizează fiecare arc exact o dată. Dacă un asemenea drum există, se spune că graful este *eulerian* sau *unicursal*.
Pentru grafurile orientate, drumurile si ciclurile sunt si ele orientate.
Definitiile si proprietătile se mentin si pentru grafurile multiple.

1. **Plicuri cu bani**. Am o pungă care conține sau o bancnotă de 1 leu, sau o bancnotă de 5 lei (cu probabilități egale pentru cele două posibilități). Adaug o bancnotă de 1 leu în pungă, asadar punga contine acum două bancnote. Punga este agitată si se extrage o bancnotă de 1 leu. Dacă se extrage si bancnota rămasă, care este sansa ca aceasta să fie de 1 leu?

Solutie: Fie A evenimentul primei extrageri (1 leu) si B evenimentul extragerii a doua (1 leu). Problema constă în a calcula probabilitatea conditionată $\Pr[B|A]$. Pentru aceasta sunt necesare $\Pr[A \cap B]$ si $\Pr[A]$.

Ambele extrageri sunt de 1 leu dacă si numai dacă bancnota pusă initial în pungă este de 1 leu. Asadar, $\Pr[A \cap B] = 1/2$. Evenimentul A se produce fie (a) dacă prima bancnotă pusă în pungă aste de 1 leu (cu probabilitatea de 1/2) si se extrage bancnota de 1 leu (cu probabilitatea 1), fie (b) prima bancnotă este de 5 lei (cu probabilitatea 1/2) si se extrage bancnota de 1 leu (cu probabilitatea 1/2). Astfel, probabilitatea $\Pr[A] = (1/2).1 + (1/2)(1/2) = 3/4$. În final

$$\Pr[B|A] = \Pr[A \cap B]/\Pr[A] = (1/2)/(3/4) = 2/3$$

2. **Un paradox al probabilităților conditionate?** Iată câteva date privind punctualitatea a două companii aeriene, A si B, pe două aeroporturi, LA si Ch. Predictibil, ambele companii au performante mai bune în LA decât în Ch din cauza congestionării traficului si din cauza conditiilor meteo.

	Compania A		Compania B	
	Zboruri	Punctuale	Zboruri	Punctuale
LA	600	534	200	188
Ch	250	176	900	685

(a) Care dintre cele două companii are sanse mai bune de a sosi la LA la vreme? Dar despre sosirea la Ch ce se poate spune?
Răspuns: Pr[linia A punctuală la LA] = 534/600 = 0,89 si Pr[linia B punctuală la LA] = 188/200 = 0,94. Linia B are sanse mai bune de punctualitate la LA.
Pr[linia A punctuală la Ch] = 176/250 = 0,704 si Pr[linia B punctuală la Ch] = 685/900 ≈ 0,761. Linia B are sanse mai bune de a ateriza punctual la Ch.

(b) Care din cele două companii au sanse mai bune de a fi punctuală în general?
Răspuns: Pr[linia A punctuală] = (534 + 176)/(600 +250) ≈ 0,835 si Pr[linia B punctuală] = (188 + 685)/(200 +900) ≈ 0,794. Asadar, linia A are sanse mai mari de a fi în general punctuală.

(c) Sunt rezultatele de la punctele (a) si (b) surprinzătoare? Explicati aparentul paradox si interpretati-l în termeni de probabilități conditionate.

Răspuns: Paradoxul constă în faptul că pe ambele aeroporturi linia B are procentaje de sosiri punctuale mai bune, dar linia A are un procentaj general mai bun. Acesta este paradoxul lui Simpson. Explicatia informală se constituie din faptul că aeroportul LA este mai bun în asigurarea aterizărilor conform orarelor si compania A are mult mai multe zboruri decât compania B spre acel aeroport. Prin urmare nu este irational ca linia A să aibe o performantă generală mai bună decât B cu toate că aceasta o întrece pe A în fiecare din cele două locuri.

Pentru o întelegere mai profundă, fie spatiul evenimentelor elementare alcătuit din toate zborurile, evenimente echiprobabile, si să definim evenimentele: AL – un zbor al companiei A cu destinatia LA, AC – un zbor A spre Ch, BL – un zbor al companiei B cu destinatia LA, BC – un zbor B spre Ch si O – un zbor cu aterizare conform orarului.

La punctul (a) s-au calculat $\Pr[O|AL]$, $\Pr[O|BL]$, $\Pr[O|AC]$ si $\Pr[O|BC]$. La punctul (b) s-au calculat $\Pr[O|(AL \cup AC)]$ si $\Pr[O|(BL \cup BC)]$. Paradoxul aparent este că

$$\Pr[O|AL] < \Pr[O|BL] \text{ si } \Pr[O|AC] < \Pr[O|BC]$$

dar

$$\Pr[O|(AL \cup AC)] > \Pr[O|(BL \cup BC)]$$

Matematic, asta se întâmplă pentru că

$$\Pr[O|(AL \cup AC)] = \frac{\Pr[O \cap (AL \cup AC)]}{\Pr[AL \cup AC]} = \frac{\Pr[(O \cap AL) \cup (O \cap AC)]}{\Pr[AL \cup AC]} =$$

$$= \frac{\Pr[O \cap AL] + \Pr[O \cap AC]}{\Pr[AL] + \Pr[AC]} = \frac{\Pr[O \mid AL]\Pr[AL] + \Pr[O \mid AC]\Pr[AC]}{\Pr[AL] + \Pr[AC]}$$

care este mai mare decât

$$\frac{\Pr[O \mid BL]\Pr[BL] + \Pr[O \mid BC]\Pr[BC]}{\Pr[BL] + \Pr[BC]}$$

pentru motivele de mai sus.

3. **Independentă**. Fie Ω un spatiu al evenimentelor elementare si fie $A, B \subseteq \Omega$ două evenimente *independente*. Fie evenimentele contrare $\overline{A} = \Omega - A$ si $\overline{B} = \Omega - B$.

Pentru rezolvarea acestei probleme poate fi utilizată ca definitie a independentei a două evenimente afirmatia: evenimentele A si B sunt *independente* dacă $\Pr[A \cap B] = \Pr[A].\Pr[B]$

(a) Demonstrati adevărul sau neadevărul afirmatiei: evenimentele $\overline{A}$ si $\overline{B}$ sunt în mod necesar independente.

Solutie: Afirmatia este adevărată.

$$\Pr[\overline{A} \cap \overline{B}] = \Pr[\overline{A \cup B}] = 1 - \Pr[A \cup B] = 1 - (\Pr[A] + \Pr[B] - \Pr[A \cap B]) =$$

$$= 1 - \Pr[A] - \Pr[B] + \Pr[A]\Pr[B] = (1 - \Pr[A])(1 - \Pr[B]) = \Pr[\overline{A}]\Pr[\overline{B}]$$

în care s-au utilizat pe rând, formulele lui De Morgan, formula de calcul al probabilitătii evenimentului contrar si al probabilitătii unei reuniuni de două evenimente, precum si definitia independentei.

(b) Demonstrati adevărul sau neadevărul afirmatiei: evenimentele A si $\overline{B}$ sunt în mod necesar independente.

Solutie: Afirmatia este adevărată.

$$\Pr[A \cap \overline{B}] = \Pr[A - (A \cap B)] = \Pr[A] - \Pr[A \cap B] =$$

$$= \Pr[A] - \Pr[A]\Pr[B] = \Pr[A](1 - \Pr[B]) = \Pr[A]\Pr[\overline{B}]$$

(c) Demonstrati adevărul sau neadevărul afirmatiei: evenimentele A si $\overline{A}$ sunt în mod necesar independente.

Solutie: Afirmatia este falsă. Pentru un eveniment oarecare, $0 < \Pr[A] < 1$, $\Pr[A \cap \overline{A}] = \Pr[\emptyset] = 0$. Dar $\Pr[A]\Pr[\overline{A}] > 0$. Asadar cele două evenimente nu sunt independente.

(d) Demonstrati adevărul sau neadevărul afirmatiei: este posibil ca $A = B$.

Solutie: Afirmatia este adevărată. Exemplu: dacă $\Pr[A] = \Pr[B] = 0$ atunci $\Pr[A \cap B] = 0 = 0 \times 0 = \Pr[A].\Pr[B]$, la fel dacă $\Pr[A] = \Pr[B] = 1$ atunci $\Pr[A \cap B] = 1 = 1 \times 1 = \Pr[A].\Pr[B]$, cu A si B independente.

4. **Corelatie**. Dacă $\Pr[A|B] > \Pr[A]$, A si B pot fi intuite ca fiind corelate pozitiv. Se poate pune întrebarea dacă această corelatie pozitivă este o relatie simetrică. Asa că, demonstrati adevărul sau falsul afirmatiei: dacă $\Pr[A|B] > \Pr[A]$, atunci în mod necesar $\Pr[B|A] > \Pr[B]$ (Se presupune că ambele probabilităti conditionate sunt bine definite, adică $\Pr[A]$ si $\Pr[B]$ nu sunt nule).

Solutie: Dacă $\Pr[A|B] > \Pr[A]$, atunci prin multiplicare cu $\Pr[B]/\Pr[A]$ se obtine $\Pr[A|B].\Pr[B]/\Pr[A] > \Pr[B]$. Dar partea din stânga inegalitătii este tocmai $\Pr[B|A]$ ($\Pr[A \cap B] = \Pr[A|B].\Pr[B] = \Pr[B|A].\Pr[A]$) ceea ce dovedeste adevărul afirmatiei.

5. **O alegere din zbor**. La aruncarea de trei ori a unei monede corecte sunt posibile opt rezultate echiprobabile: SSS, SSR, SRS, SRR, RSS, RSR, RRS si RRR. Doi studenti joacă un joc pe bază de aruncarea monedei. Jucătorul A alege o triplă din cele enumerate; jucătorul B alege una diferită. Moneda este aruncată repetat până când apare una din triplele alese care si câstigă jocul. De exemplu dacă jucătorul A alege SSR si B alege SRS si aruncările au ca rezultat RSSSR, jucătorul A câstigă.

   Completati un tabel cu cele mai potrivite alegeri ale lui B pentru fiecare alegere a lui A. "Cele mai potrivite" înseamnă alegerile care fac pentru B probabilitatea de câstig maximă (se sugerează utilizarea unui mic program de calcul).

   Explicati de ce sansele de câstig sunt asa de dezechilibrate în favoarea unui jucător.

Alegerea lui A	Alegerea lui B (cea mai bună)	Probabilitatea de câstig pentru B
SSS	RSS	7/8
SSR	RSS	3/4
SRS	SSR	2/3
SRR	SSR	2/3
RSS	RRS	2/3
RSR	RRS	2/3
RRS	SRR	3/4
RRR	SRR	7/8

Solutie: Deorece B cunoaste deja alegerea făcută de A, el poate să-si ajusteze alegerea pe baza alegerii lui A. Dacă A alege SSS atunci B alege RSS, A poate câstiga numai dacă prima aruncare este S; dacă B obtine RRR în primele trei aruncări el nu mai poate pierde. Dacă A alege SSR, el poate câstiga numai dacă primele două aruncări sunt SS.
Se admite că A alege SRS si B alege SSR. Fie p probabilitatea ca A să câstige. Să detaliem posibilitătile de succesiune a rezultatelor după ce apare prima stemă (S):

- Dacă secventa este R*SS (adică un număr de reversuri (R) urmate de SS) atunci A nu poate câstiga.
- Dacă secventa este R*SRS, A câstigă sigur.
- Dacă secventa este R*SRR avem o situatie de pornire identică cu prima si probabilitatea ca A să câstige de aici încolo este p.
  Acestea sunt toate cazurile. Ultimele două au probabilitatea 1/4. Asta înseamnă că
  $$p = \frac{1}{4} + \frac{1}{4}p$$
  cu solutia $p = 1/3$. Celelate linii din tabel care contin 2/3 sunt similare.
  Calea rapidă pentru B de a alege combinatia ce mai probabilă: dacă A alege $x_1x_2x_3$, B trebuie să aleagă $\bar{x}_1x_2x_3$ unde bara de deasupra indică contrarul unui rezultat ($\overline{S} = R$, $\overline{R} = S$).

6. **Cum se învinge căldura**. Este o zi fierbinte de vară. Trei copii A, B si C decid a se răcori printr-un duel în trei moduri cu baloane cu apă. Încep prin a trage la sorti cine aruncă primul, cine al doilea si cine al treilea, apoi îsi ocupă locurile în vârfurile unui triunghi echilateral. Ei cad de acord să arunce un singur balon cu apă o dată, pe rând si să continue în aceeasi ordine ciclică până când doi din ei sunt uzi. Fiecare jucător poate arunca la rândul lui în oricare din deilalti doi sau poate renunta (pasa) la turul lui. Se presupune că fiecare combatant are o provizie practic infinită de "munitie", un balon explodează odată ce a atins tinta udând-o, dacă un balon ratează tinta el explodează suficient de departe ca să nu ude pe nimeni.

Toti trei stiu că A loveste tinta totdeauna, B este imprecis în proportie de 20%, iar C loveste cu 50% sanse. Strategia lui A este să arunce spre B dacă n-a fot încă lovit si spre C dacă B a fost lovit. B va arunca spre A pentru a evita lovirea lui de către A. C nu aruncă spre niciunul la rândul lui până când unul din A si B este lovit; el face prima aruncare spre supravietuitor ceea ce i-ar putea crea un avantaj.

Astfel, iată un scenariu-exemplu cu C primul la aruncare, apoi B, apoi A. C pasează, B aruncă spre A dar ratează. A lichidează pe B. C aruncă norocos spre A si câstigă.

Care sunt sansele de supravietuire pentru fiecare, A, B si C? (Explicati clar solutia otinută).

Solutie: Sunt 6 ordini de bătaie posibile. Sunt examinate pe rând.

ABC: A loveste pe B cu probabilitate 1. Apoi C îsi îndreaptă tirul spre A. Dacă loveste, câstigă. Dacă ratează, A loveste sigur pe C si câstigă. Asadar, probabilitătile de supravietuire pentru A, B, C sunt respectiv 1/2, 0, 1/2.

ACB: La fel ca în cazul ABC.

BAC: B tinteste asupra lui A. Dacă loveste (80% probabilitate) atunci B si C rămân într-un duel cu C la prima lovitură. C câstigă dacă îl loveste pe B în încercarea a n-a, după ce si C, si B ratează primele $n - 1$ încercări. Probabilitatea ca participantul C să îl lovească pe B în încercarea a n-a este $(0,5.0,2)^{n-1}.(0,5) = (0,5)(0,1)^{n-1}$. Astfel, probabilitatea de câstig în acest duel pentru C este

$$\sum_{n=1}^{\infty} (0,5)(0,1)^{n-1} = \frac{0,5}{1-0,1} = \frac{5}{9}$$

B are sansa a doua în acest duel: $1 - (5/9) = 4/9$.

Dacă B ratează prima aruncare (20% probabilitate), atunci A îl loveste sigur în etapa următoare. Lui C îi rămân 50% sanse de supravietuire. Probabilitătile de supravietuire sunt:

A: $0,8.0 + 0,2.0,5 = 0,1$

B: $0,8.(4/9) + 0,2.0 = 16/45$

C: $0,8.(5/9) + 0,2.0,5 = 49/90$

BCA: Din nou B tinteste la A. Dacă îl loveste atunci C si B rămân într-un duel ca în cazul precedent. Dacă B ratează lovitura asupra lui A, C pasează si A loveste (cu sigurantă) pe B. C si A se bat ca în cazul precedent. Probabilitătile sunt ca si în cazul BAC.

CAB: Este la fel cu ABC deoarece C pasează în primul tur.

CBA: Este la fel cu BAC deoarece C pasează în primul tur.

Se pot calcula acum probabilitătile generale de supravietuire:

A: $(1/6)(1/2 + 1/2 + 1/10 + 1/10 + 1/2 + 1/10) = 3/10$

B: $(1/6)(0 + 0 + 16/45 + 16/45 + 0 + 16/45) = 8/45$

C: $(1/6)(1/2 + 1/2 + 49/90 + 49/90 + 1/2 + 49/90) = 47/90$

1. **Unităti repetate de patru ori**. Spunem că un sir de biti are *k secvente de patru unităti binare* dacă sunt *k* pozitii în care apar patru unităti consecutive. De pildă, sirul 1001111100 are de două ori secventa de patru biti consecutivi. Care este numărul mediu de secvente de patru unităti binare într-un sir aleator de *n* biti, când $n > 3$ si toate sirurile de biti de lungime *n* sunt egal probabile? Justificati răspunsul.
   Solutie: Fie *S* variabila aleatoare care desemnează un sir de *n* biti la întâmplare si *X* numărul de pozitii de început al unor unităti binare cvadruple din *S*. Se defineste variabia aleatoare X_i astfel:
   $$X_i = \begin{cases} 1 & \text{daca } S \text{ are patru de 1 la pozitia } i \\ 0 & \text{altminteri} \end{cases}$$
   De retinut că $\Pr[X_i = 1] = 1/16$, astfel încât $E[X_i] = 1/16$. Acum, $X = X_1 + \ldots + X_{n-3}$. Din liniaritatea mediei, $E[X] = E[X_1] + \ldots + E[X_{n-3}] = (n - 3)/16$.

2. **Mize bine făcute**. Doi jucători A si B mizează fiecare câte 32 de pistoli pe o sansă de trei puncte, câstigătorul ia tot. Jocul este jucat în runde; la fiecare rundă unul din jucători câtigă un punct celălalt nu câstigă nimic. În mod normal, primul jucător care acumulează 3 puncte ia toti cei 64 de pistoli. Dar în timpul jocului se pune ploaia, la un punct unde A are 2 puncte, B are 1 punct. A si B au de împărtit banii. Cum o fac?
   Solutie: Trebuie admis că A si B sunt pe măsură, la fiecare rundă sansele de câstig sunt 50-50%. Se admite totodată că partea lui A ar trebui să fie proportională cu maximum a ceea ce el ar vrea să plătească pentru a continua jocul de la punctul în care s-a întrerupt. Stiinta economiei spune că maximum ce ar trebui să plătească A este exact media conditionată a câstigurilor lui, mai specific, câstigurile lui dacă jocul ar continua din punctul opririi (este aici o doză de simplificare; admitem că functia de utilitate a lui A este functia identitate; această presupunere poate fi în practică perfectă sau imperfectă; este însă rezonabilă ca primă aproximare pentru jocurile cu miză mică). Aceeasi problemă se formulează si pentru B.
   Calculati un mod corect de a distribui cei 64 de pistoli utilizând această notiune de corectitudine. Câti pistoli primeste A, câti B?
   Miezul problemei constă în a calcula sansa lui A de a câstiga dacă jocul ar fi continuat. Media câstigului este exact de 64 de ori acea sansă. Să calculăm probabilitatea.
   Sansa lui A de a câstiga în rundul următor este de 1/2 deoarece ei sunt deopotrivă în joc. În plus, sansa lui B să câstige punctul următor este 1/2 caz în care jocul continuă. Sansa lui A de a câstiga este din nou 1/2. De aici încolo, desemnarea unui câstigător este inevitabilă, cineva, A sau B trebuie

în mod necesar să acumuleze trei puncte. Astfel A are sansa de câstig de 1/2 + (1/2)(1/2) = 3/4.

Se poate rationa si altfel. Este limpede că jocul se termină în mod necesar în următoarele două runde. Rezultatele posibile sunt AA, AB, BA si BB care sunt toate la fel de probabile. Deoarece A câstigă în trei din patru cazuri, sansa lui de a câstiga este de 3/4.

În concluzie, la întrerupere A ia 48 de pistoli, B numai 16.

3. **O amestecare defectuoasă**. Se consideră o metodă rea de amestecare (adică de de permutare aleatoare) a unui pachet A de 52 de elemente.
   (a) Se initializează un pachet A_{nou} pentru a contine 52 de indicatori "vizi" (pluralul de la vid)
   (b) Pentru $k = 0$ la 51:
      i. Se generează repetat un întreg j între 0 si 51 până când $A[j]$ nu este vid
      ii. Se copiază $A[j]$ în $A_{\text{nou}}[k]$ si se pune $A[j]$ ca fiind "vid"
   (c) Se copiază A_{nou}, care contine acum elementele amestecate, în A.

Determinati numărul mediu de întregi aleatori j care sunt necesari pentru a produce $A_{\text{nou}}[k]$.

Solutia 1. Fie X variabila aleatoare care reprezintă numărul de întregi aleatori j generati pentru a produce $A_{\text{nou}}[k]$. X se poate exprima ca suma unui număr infinit de variabile aleatoare indicator X_i cu $X_i = 0$ dacă este produs un $A_{\text{nou}}[k]$ în mai putin de i generări de j si $X_i = 1$ dacă sunt necesare cel putin i generări. Dacă sunt necesare n generări, atunci $X_i = 1$ pentru $i \le n$ si $X_i = 0$ pentru $i > n$.

Probabilitatea de a nu reusi producerea unui $A_{\text{nou}}[k]$ în primele $i - 1$ generări este $(k/52)^{i-1}$ deoarece sunt k elemente vide în A. Asadar, valoarea medie $E[X_i]$ este $(k/52)^{i-1}$. Media generală a lui X este

$$E[X] = \sum_{i=1}^{\infty} E[X_i] = \sum_{i=1}^{\infty} \left(\frac{k}{52}\right)^{i-1} = \frac{1}{1 - \frac{k}{52}} = \frac{52}{52 - k}$$

Solutia 2. Formula pentru numărul mediu de generări necesare pentru a produce un $A_{\text{nou}}[k]$ poate fi utilizată direct. Probabilitatea de a produce un $A_{\text{nou}}[k]$ în exact i generări este $(k/52)^{i-1}(52 - k)/52$ deoarece se pot genera $i - 1$ elemente vide $A[j]$ înainte de a genera un element nevid la încercarea a i-a. Valoarea medie a lui X (definită similar ca la solutia 1) este

$$E[X] = \sum_{i=1}^{\infty} i\left(\frac{k}{52}\right)^{i-1} \frac{52 - k}{52} = \frac{52 - k}{52} \sum_{i=1}^{\infty} i\left(\frac{k}{52}\right)^{i-1}$$

Dacă se pune

$$S = \sum_{i=1}^{\infty} i\left(\frac{k}{52}\right)^{i-1} = \sum_{i=0}^{\infty} (i+1)\left(\frac{k}{52}\right)^{i}$$

atunci

$$\frac{k}{52}S = \sum_{i=1}^{\infty} i\left(\frac{k}{52}\right)^i$$

si

$$S - \frac{k}{52}S = \sum_{i=0}^{\infty}\left(\frac{k}{52}\right)^i = \frac{1}{1-\frac{k}{52}} = \frac{52}{52-k}$$

ceea ce înseamnă că

$$S = \left(\frac{52}{52-k}\right)^2$$

În consecintă

$$E[X] = \frac{52-k}{52}.S = \frac{52}{52-k}$$

4. **Analiza liftului**. Două lifturi se deplasează continuu de la etajul cel mai de sus la subsol si înapoi. La fiecare etaj este un singur buton care dacă este actionat produce oprirea următorului lift care trece pe la acel etaj, indiferent în ce directie se deplasează.
   Un matematician M de la etajul 8 decide să facă o analiză a sistemului de lifturi. Clădirea are 10 etaje, parter si subsol. Initial M crede că probabilitatea ca un lift să meargă în sus ar fi 9/11 deoarece sunt două etaje deasupra etajului 8 si distanta totală de sus până jos este de 11 etaje. Dar observatiile sugerează că probabilitatea ca liftul care răspunde să meargă în sus este în realitate apropiată de 2/3.
   Stabiliti probabilitatea reală ca un lift care răspunde la o chemare de la etajul 8 să meargă în sus si explicati cum ati obtinut rezultatul. Presupuneti că sunt două lifturi independente care functionează conform explicatiilor.
   Solutie: Să denumim lifturile A si B. Să considerăm pozitiile acestora la momentul chemării. Locatiile fiecăruia sunt uniform distribuite pe puturile lor si sunt independente. Sunt trei scenarii în care primul lift care atinge etajul 8 să meargă în sus:

- Ambele lifturi sunt la momentul chemării sub etajul 8. Probabilitatea ca lifturile să fie pozitionate astfel este $(9/11)^2 = 81/121$. Următorul lift care atinge etajul 8 va merge cu sigurantă în sus.
- Liftul A este deasupra etajului 8 si B este între etajele 4 si 8 si merge în sus. Probabilitatea acestor pozitii pentru A si B este $(2/11)(4/11)(1/2) = 4/121$. În cazul acesta, fiecare lift trebuie să parcurgă până la 4 etaje pentru a ajunge la etajul 8 (dacă A tocmai a părăsit etajul 8 mergând în sus, el parcurge două etaje în sus până la 10 si două etaje în jos până la 8. Astfel, fiecare lift are aceeasi probabilitate de a ajunge la etajul 8 si următorul lift va merge în sus dacă si numai dacă acela este B. Asta înseamnă că dacă A si B sunt în pozitiile descrise, jumătate din timp următorul lift va merge în sus.

În total, acest caz contribuie cu (4/121)(1/2) = 2/121 la probabilitatea ca următorul lift să meargă în sus.

- Pozitiile lui A si B inversate.
  În celelalte situatii în care fie (a) ambele lifturi sunt deasupra etajului 8 fie (b) unul este deasupra etajului 8 si celălalt ori merge în jos or este sub etajul 4, următorul lift care ajunge la etajul 8 va merge în jos.
  Probabilitatea totală ca următorul lift să meargă în sus este

$$\frac{81}{121} + \frac{2}{121} + \frac{2}{121} = \frac{85}{121}$$

5. **O martingală**. Se consideră un *joc corect* într-un cazino: la fiecare joc mizezi un număr de S lei si pierzi sau câstigi cu probabilitatea 1/2 (jocurile sunt independente); dacă câstigi, iei miza plus S lei, dacă pierzi, pierzi miza.
   (a) Care este numărul de jocuri mediu până la primul câstig (inclusiv jocul câstigător)?
   Răspuns: La fiecare rundă sansa de a câstiga este $p = 1/2$. Este evidentă echivalenta problemei cu aceea a aruncării unei monede corecte până la aparitia primei steme, problemă prezentată în/la curs. Valoarea medie a numărului de runde de joc până la primul câstig (inclusiv) este $1/p = 1/(1/2) = 2$.
   (b) Strategia de joc următoare, numită si "martingală", a fost foarte populară în cazinourile secolului 18: la primul joc se miza 1 unitate, la jocul al doilea se miza 2, la al treilea 4, la jocul k se miza 2^{k-1}, oprire si părăsirea cazinoului la primul câstig!
   Arătati că dacă urmati strategia martingalei si aveti resurse bănesti nelimitate, la plecarea din cazino veti fi cu sigurantă mai bogat cu 1 unitate. De aceea, probabil, strategia a fost interzisă în multe din cazinourile moderne.
   Solutie: Dacă pierdeti în runda i-a, pierderea este de 2^{i-1} în acea rundă; dacă câstigati, câstigul este de 2^{i-1}. Să admitem că primele $k - 1$ ture sunt cu pierdere dar runda k este câstigătoare. Câstigul net este

$$-1 - 2 - 4 - \ldots - 2^{k-2} + 2^{k-1} = -\frac{2^{k-1} - 1}{2 - 1} + 2^{k-1} = 1$$

   oricare ar fi k. Dar k poate fi oricât de mare, deci resursa de bani trebuie să fie la fel, oricât de mare.
   (c) C si D vin în cazino cu 10 lei, respectiv cu 1.000.000 de lei. Amândoi joacă după strategia martingalei cu adaosul că ei trebuie să părăsească cazinoul si înainte de primul câstig dacă li se termină banii. Care este probabilitatea ca unul sau altul să câstige?
   Răspuns: Fie p sansa lui C de a câstiga. Este usor de calculat $(1 - p)$, riscul lui ca după 3 jocuri să piardă (în total 7 lei): $1/2^3 = 1/8$. Asadar $p = 7/8$.
   Un calcul similar pentru D: pentru a pierde el trebuie să piardă toate rundele dinaintea rămânerii fără bani. Dacă asta se întâmplă în runda k,

pierderea este de $2^k - 1$. Deoarece are 1.000.000 de lei, k trebuie să fie 19, etapă la care a pierdut deja 524.288 – 1 lei. Asadar, riscul de a se produce asa ceva este de 1/219 si probabilitatea de câstig este 0,99999809.

Comentariu: Ar putea părea tentant jocul la o asemenea probabilitate de câstig vecină cu siguranta. Din nefericire câstigul este de numai 1 leu. Dacă D pierde, pierde o mică avere. Asadar, atentie!

1. **Independență**.

(a) Arătați că pentru variabilele aleatoare independente X, Y avem $E[XY] = E[X]E[Y]$. *Indicatie*: Arătati mai întâi, cu grijă!, că si dacă cele două variabile *nu* sunt independente, $E[XY] = \Sigma_a \Sigma_b ab \Pr[X = a \wedge Y = b]$.

Solutie: Arătăm mai întâi că $E[XY] = \Sigma_a \Sigma_b ab \Pr[X = a \wedge Y = b]$:

$$E[XY] = \sum_c c \Pr[XY = c] = \sum_c c \sum_{a,b|ab=c} \Pr[X = a \wedge Y = b] =$$

$$= \sum_c \sum_{a,b|ab=c} ab \Pr[X = a \wedge Y = b] = \sum_{a,b} ab \Pr[X = a \wedge Y = b] =$$

$$= \sum_a \sum_b ab \Pr[X = a \wedge Y = b]$$

succesiune de egalități care tine seamă de definitia valorii medii, de descompunerea evenimentului $XY = c$ în submultimi disjuncte, de aparitia perechii (a, b) o singură dată si numai o dată.

Apoi folosim rezultatul acesta pentru a arăta că $E[XY] = E[X] \times E[Y]$:

$$E[XY] = \sum_a \sum_b ab \Pr[X = a \wedge Y = b] = \sum_a \sum_b ab \Pr[X = a] \Pr[Y = b] =$$

$$= \sum_a a \Pr[X = a] \sum_b b \Pr[X = b] = \left(\sum_a a \Pr[X = a] \right) \left(\sum_b b \Pr[X = b] \right) = E[X]E[Y]$$

ceea ce tine seama de independenta lui X de Y, de distributivitatea adunării si de definitiile mediilor $E[X]$ si $E[Y]$.

(b) Dati un exemplu simplu care să arate că concluzia punctului anterior nu este în mod necesar adevărată dacă X si Y nu sunt independente.

Solutie: Se presupune că X este fie 0, fie 2 cu probabilități egale si $Y = X$ totdeauna. În acest caz $E[XY] = E[X^2] = 2$ dar $E[X] \times E[Y] = (E[X])^2 = 1^2 = 1$, astfel încât concluzia de mai sus poate fi falsă în general (si, desigur, în cazul acesta X si Y nu sunt independente).

2. **Cele 3407 voturi**. După alegerile prezidentiale americane din 2000, multi oameni au clamat că cele 3407 voturi pentru candidatul Pat Buchanan în comitatul Palm Beach sunt înalt semnificative statistic si, de aceea, dubioase sub aspectul validității. În această problemă se verifică aici această pretentie din punct de vedere statistic.

   Procentele obtinute de fiecare candidat prezidential în toată Florida au fost:

Gore	Bush	Buchanan	Nader	Browne	Altii
48,8%	48,9%	0,3%	1,6%	0,3%	0,1%

În comitatul Palm Beach, voturile (înainte de a începe renumărarea) au fost:

Gore	Bush	Buchanan	Nader	Browne	Altii
268945	152846	3407	5564	743	781

cu un total de 432286 de voturi.

Pentru a modela probabilistic această situatie, este necesară formularea câtorva ipoteze. Să modelăm executarea votului de fiecare votant din numitul comitat ca o variabilă aleatoare X_i, unde X_i ia pentru fiecare din cele sase valori posibile (cinci candidati si "altii") cu probabilitătile corespunzătoare procentajelor pe Florida (de plidă, $\Pr[X_i = \text{Gore}] = 0{,}488$). Sunt în total 432286 votanti si voturile lor sunt considerate mutual indepndente. Fie variabila aleatoare B cea care exprimă toate voturile în favoarea lui Buchanan în comitatul Palm Beach (numărul de votanti i pentru care X_i = Buchanan).

(a) Calculati media $E[B]$ si dispersia $Var[B]$.

Solutie: Fie B_i variabila aleatoare care reprezintă votul votantului i pentru Buchanan, în detaliu $B_i = 1$ dacă si numai dacă X_i = Buchanan. De observat că variabilele B_i sunt independente si identic distribuite, cu media $E[B_i] = 0{,}003$ si $Var[B_i] = 0{,}003\times(1 - 0{,}003) = 0{,}002991$. În plus, prin liniaritatea mediei si independentă stabilim că $E[B] = \sum_{i=1}^{n} E[B_i] = 432286\times 0{,}003 \approx 1297$ si $Var[B] = \sum_{i=1}^{n} Var[B_i] = 432286\times 0{,}002991 \approx 1293$.

(b) Folositi inegalitatea lui Cebîsev pentru a calcula limita de sus b pentru probabilitatea ca Buchanan să primească 3407 voturi, adică găsiti un număr b astfel încât $\Pr[B \geq 3407] \leq b$. Pe baza aceasta apreciati dacă votul pentru Buchanan este semnificativ.

Solutie: Inegalitatea lui Cebîsev promite că $\Pr[|B - E[B]| \geq a] \leq Var[B]/a^2$. În cazul de fată, cu $E[B] = 1297$ si $Var[B] = 1293$, si trebuie luat $a = 2110$, ceea ce duce la $\Pr[|B - 1297| \geq 2110] \leq 1293/2110^2 \approx 0{,}0003$. Acum facem observatia că inegalitatea $|B - 1297| < 2110$ este echivalentă conditiei $-813 < B < 3407$ si deoarece B este nenegativ, găsim că $\Pr[B > 3407] \leq 0{,}0003$ (aproximativ), astfel încât putem lua $b \approx 0{,}0003$. Cu alte cuvinte, obtinerea de către Buchanan a 3407 voturi în comitatul Palm Beach pare, pe baza acestui model simplu, putin probabilă datorată întâmplării.

(c) Presupuneti acum că limita b stabilită la punctul anterior este de fapt exactă, adică "$\leq$" devine "$=$" (de fapt valoarea adevărată a acestei probabilităti este putin mai mică decât b). Presupuneti totodată că toate cele 67 de comitate ale Floridei au acelasi număr de votanti ca si comitatul Palm Beach si că toate se comportă independent potrivit aceluiasi model statistic adoptat pentru comitatul Palm Beach. Care este probabilitatea ca în cel putin un

comitat, Buchanan să primească cel putin 3407 voturi? Cum ar afecta asta judecata voastră despre semnificatia scorului din comitatul Palm Beach?

Solutie: Fie p_j probabilitatea ca un comitat, al j-lea să nu înregistreze 3407 voturi pentru Buchanan. De la punctul (b) avem $p_j = 1 - b \approx 0{,}9997$. Se observă că probabilitatea ca nici unul din comitate să nu producă 3407 voturi pentru Buchanan este $p_1 \times \ldots \times p_{67}$ deoarece votantii din oricare comitat se comportă independent. Astfel, probabilitatea ca Buchanan să nu primească 3407 voturi în vreunul din toate comitatele este $(0{,}9997)^{67} \approx 0{,}98$. În consecintă, probabilitatea ca Buchanan să primească cel putin 3407 voturi în unele comitate este de circa $1 - 0{,}98 \approx 0{,}02$. Altfel spus, acest fapt este foarte putin probabil să se întâmple prin hazard.

(d) Modelul nostru presupune că toti votantii se comportă ca "votanti swing" în sensul că ei sunt nedecisi când merg la votare si sfârsesc prin a lua o decizie aleatoare. Un model mai realist ar presupune că numai o fractie (să spunem, 20%) din votanti sunt în categoria "swing", ceilalti fiind deja decisi. Presupunem că 80% din votantii din comitatul Palm Beach votează determinist, potrivit cu proportiile proprii statului Florida în întregime si numai 20% se comportă aleator cum s-a spus mai devreme. Cu acest model, limita b de la punctul (b) creste, scade sau rămâne la fel? Justificati răspunsul.

Solutie: În modelul modificat, b creste deoarece o fractie încă mai mare din alegătorii "aleatori" ar fi votat pentru Buchanan pentru a produce o proportie atât de neuzual de mare de voturi pentru Buchanan.

3. **Găsiti jokerul**.

(a) Luati un pachet de cărti obisnuit de 52 de cărti si adăugati un joker. Amestecati si întoarceti câte o carte până apare jokerul. Câte cărti în medie trebuie întoarse până apare jokerul?

Solutie: Fie variabila aleatoare X cea care marchează pozitia jokerului în pachetul de 53 de cărti: este o variabilă uniform distribuită pe multimea $\{1, 2, \ldots, 53\}$. Media ei este $E[X] = (1 + 2 + \ldots + 53)/53 = 27$. Asadar, trebuie întoarse în medie peste 27 de cărti pentru a vedea jokerul (presupunând că jokerul se numără printre aceste 27; dacă jokerul se exclude atunci sunt necesare 26).

(b) Luati acum pachetul de 52 de cărti si adăugati doi jokeri, amestecati si întoarceti carte cu carte până apare primul joker. Câte cărti în medie trebuie întoarse până apare primul joker?

Solutia 1: Punem perechea ordonată (Y, Z) să reprezinte pozitia în pachet a primului si a celui de al doilea joker, respectiv. Perechea aleatoare (Y, Z) este uniform distribuită pe multimea perechilor (i, j) cu $1 \le i < j \le 54$. Sunt C_{54}^2 astfel de perechi si toate sunt egal probabile. Mai mult, sunt 53 de perechi de forma (1, *) cu * o carte oarecare, diferită de prima, 52 de perechi de forma (2, *) si, în general $54 - i$ perechi de forma $(i, *)$. Astfel, $\Pr[Y = i] = (54 - i)/C_{54}^2$. În consecintă

$$E[Y] = \sum_{i=1}^{53} i(54-i)/C_{54}^2 = \frac{1}{1431}\left(53\sum_{i=1}^{53} C_i^1 - 2\sum_{i=1}^{53} C_i^2\right) =$$

$$= \frac{1}{1431}\left(53C_{54}^2 - 2C_{54}^3\right) = \frac{75843-49608}{1431} = 18\frac{1}{3}$$

Asadar, în medie 18,333 cărti trebuie întoarse pentru a vedea primul joker (numărând si jokerul) dacă în pachet sunt introdusi 2 jokeri.

Solutia 2: Fie X numărul de cărti întoarse până la si încluzând primul joker. Atunci

$$X = X_1 + X_2 + \ldots + X_{53}$$

Cu $X_i = 1$ dacă pozitia primului joker este $\geq i$ si $X_i = 0$ altminteri. Media este

$$E[X] = \sum_{i=1}^{53} E[X_i]$$

Fie J_i evenimentul care constă în "cartea i este un joker". Cu putin spirit de observatie putem scrie $\Pr[X_1 = 1] = 1$, $\Pr[X_2 = 1] = \Pr[\neg J_1] = 52/54$, $\Pr[X_3 = 1] = \Pr[\neg J_1 \wedge \neg J_2] = \Pr[\neg J_1]\Pr[\neg J_2|\neg J_1] = (52/54)(51/53)$ si, în general, $\Pr[X_i = 1] = (55 - i)(54 - i)/54/53$ prin telescoparea produsului. În consecintă

$$E[X] = \sum_{i=1}^{53} \frac{(55-i)(54-i)}{54.53} = \frac{1}{1431}\sum_{i=1}^{53} C_{55-i}^2 =$$

$$= \frac{1}{1431}\sum_{j=1}^{54} C_j^2 = \frac{1}{1431}C_{55}^3 = \frac{26235}{1431} = 18\frac{1}{3}$$

(c) Verificati rezultatele prin scrierea unui program de calcul. Executati-l de un număr mare de ori (de pildă de un milion de ori) si calculati numărul mediu de cărti necesar a fi întoarse până la aparitia primului joker. Ce ati obtinut?

Solutie: S-au obtinut 27,0044 si 18,3362 ceea ce arată destul de aproape de valorile teoretice.

4. **Variabile aleatoare modulo p**. Fie variabilele aleatoare X si Y distribuite independent si uniform în multimea $\{0, 1, \ldots, p-1\}$, cu $p > 2$ si prim.

(a) Care este media $E[X]$?

Solutie: Se calculează direct utilizând definitia mediei:

$$E[X] = 0.\Pr[X = 0] + 1.\Pr[X = 1] + \ldots + (p-1).\Pr[X = p-1] =$$
$$= 0/p + 1/p + 2/p + \ldots + (p-1)/p = [1 + 2 + \ldots + (p-1)]/p =$$
$$= [p(p-1)/2]/p = (p-1)/2$$

(b) Fie $S = (X + Y) \bmod p$ si $T = XY \bmod p$. Care sunt distributiile lui S si T?

Solutie: Pentru orice x si y din multimea $\{0, 1, \ldots, p-1\}$, $\Pr[X = x \wedge Y = y] = 1/p^2$ (deoarece $\Pr[X = x] = 1/p$, $\Pr[Y = y] = 1/p$ si X si Y sunt variabile aleatoare independente).

Fie $s \in \{0, 1, \ldots, p-1\}$. Avem

$$\Pr[S = s] =$$

$$= \sum_{x=0}^{p-1} \Pr[S = s \wedge X = x] = \sum_{x=0}^{p-1} \Pr[X = x \wedge Y \equiv s - x \pmod p] = \sum_{x=0}^{p-1} \frac{1}{p^2} = \frac{1}{p}$$

Pentru a calcula $\Pr[T = t]$ sunt de analizat două cazuri: $t = 0$ si $t \neq 0$. Pentru $t = 0$, $XY \equiv 0 \pmod p$ dacă sau numai dacă $X = 0$ sau $Y = 0$ (sau ambele). Asadar:

$$\Pr[T = 0] = \Pr[X = 0 \text{ sau } (X \neq 0 \wedge Y = 0)] =$$
$$= \Pr[X = 0] + \Pr[X \neq 0 \wedge Y = 0] = \Pr[X = 0] + \Pr[X \neq 0] \times \Pr[Y = 0] =$$
$$= (1/p) + [(p - 1)/p](1/p) = (2p - 1)/p^2$$

uzând de proprietatea reuniunii de evenimente disjuncte si de independenta celor două variabile X si Y.

Pentru $t \neq 0$

$$\Pr[T = t] =$$
$$= \sum_{x=1}^{p-1} \Pr[T = t \wedge X = x] = \sum_{x=1}^{p-1} \Pr[X = x \wedge Y = tx^{-1} \pmod p] = \sum_{x=1}^{p-1} \frac{1}{p^2} = \frac{p-1}{p^2}$$

Pentru sigurantă, verificăm sumele probabilităților calculate. Acestea trebuie să fie egale cu unitatea. Pentru variabila sumă S cele p rezultate au fiecare probabilitatea $1/p$: suma este 1. Pentru variabila produs T avem rezultatul 0 cu probabilitatea $(2p - 1)/p^2$ si celelalte $p - 1$ rezultate cu probabilitatea $(p - 1)/p^2$. Însumate acestea dau

$$\frac{2p-1}{p^2} + (p-1)\frac{p-1}{p^2} = \frac{2p-1+(p-1)^2}{p^2} = 1$$

(c) Care sunt mediile $E[S]$ si $E[T]$?

Solutie: Prin utilizarea valorilor de la punctul (b) în definitia mediei se obtin:

$$E[S] = \sum_{s=0}^{p-1} s \Pr[S = s] = \frac{1}{p}\sum_{s=0}^{p-1} s = \frac{1}{p}\frac{p(p-1)}{2} = \frac{p-1}{2}$$

$$E[T] = \sum_{t=0}^{p-1} t \Pr[T = t] = \frac{2p-1}{p^2}.0 + \frac{p-1}{p^2}\sum_{t=0}^{p-1} t = \frac{p-1}{p^2}\frac{p(p-1)}{2} = \frac{(p-1)^2}{2p}$$

(d) Prin liniaritatea mediei, este de asteptat ca $E[S] = (E[X] + E[Y]) \bmod p$. Explicati de ce această egalitate nu este adevărată în aceste împrejurări si de ce valoarea $E[S]$ de la punctul (c) nu contrazice liniaritatea mediei?

Solutie: Liniaritatea mediei spune numai că $E[X + Y] = (E[X] + E[Y])$, nu si că $E[S] \equiv (E[X] + E[Y]) \pmod p$.

Pentru a vedea de ce acesta este un caz deosebit, să încercăm a dovedi un gen de "liniaritate a mediei" modulo p si să vedem de ce aceasta este eronată. Prin liniaritatea uzuală a mediei, $E[X + Y] = (E[X] + E[Y])$. Aceasta implică $E[X + Y] \equiv (E[X] + E[Y]) \pmod p$. Asadar, a dovedi că $E[(X + Y) \bmod p] \equiv (E[X] + E[Y]) \pmod p$ este echivalent cu a dovedi că $E[(X + Y) \bmod p] \equiv E[X + Y] \pmod p$.

Avem $X + Y = (X + Y) \bmod p + \left\lfloor \frac{X+Y}{p} \right\rfloor p$. Prin liniaritatea mediei

$$E[X + Y] = E[(X + Y) \bmod p] + p.E\left[\left\lfloor \frac{X+Y}{p} \right\rfloor\right]$$

Dacă $E\left[\left\lfloor \frac{X+Y}{p} \right\rfloor\right]$ ar fi un întreg asta ar însemna că $E[(X + Y) \bmod p] \equiv E[X + Y] \pmod p$. Dar $E\left[\left\lfloor \frac{X+Y}{p} \right\rfloor\right]$ nu trebuie să fie neapărat un întreg, motiv pentru care relatia aceasta nu are loc si "liniaritatea mediei" modulo p nu este adevărată.

Notă: Este destul de usor a arătat că $E\left[\left\lfloor \frac{X+Y}{p} \right\rfloor\right]$ nu este un întreg. Deoarece $0 \le X \le p - 1$ si $0 \le Y \le p - 1$, $0 \le X + Y \le 2p - 2$ si $\left\lfloor \frac{X+Y}{p} \right\rfloor$ este fie 0, fie 1. Asadar, $E\left[\left\lfloor \frac{X+Y}{p} \right\rfloor\right] = \Pr\left[\left\lfloor \frac{X+Y}{p} \right\rfloor = 1\right]$ si această probabilitate este clar pozitivă (deoarece $\left\lfloor \frac{X+Y}{p} \right\rfloor$ poate fi 1) si mai mică decât 1 (deoarece $\left\lfloor \frac{X+Y}{p} \right\rfloor$ poate fi 0).

(e) Deoarece X si Y sunt independente, ne-am putea astepta ca $E[T] = E[X]E[\mathrm{Y}] \bmod p$. Se mentine egalitatea în acest caz? Explicati de ce da sau de ce nu.

Solutie: Analog într-un fel cu punctul anterior, independenta variabilelor X si Y implică $E[XY] = E[X]E[Y]$ si nu $E[XY \bmod p] \equiv E[X]E[Y] \pmod p$.

Din nou, avem $\mathrm{XY} = (\mathrm{XY}) \bmod p + \left\lfloor \frac{XY}{p} \right\rfloor p$ si $E[XY] = E[(XY) \bmod p] + p \cdot E\left[\left\lfloor \frac{XY}{p} \right\rfloor\right]$. Dacă $E\left[\left\lfloor \frac{XY}{p} \right\rfloor\right]$ ar fi un întreg, atunci $E[XY \bmod p] \equiv E[XY] \pmod p$ (si în consecintă $E[XY \bmod p] \equiv E[X]E[Y] \pmod p$). Dar $E\left[\left\lfloor \frac{XY}{p} \right\rfloor\right]$ nu trebuie în mod necesar să fie întreg si de aici $E[X]E[Y] \pmod p$ n-are de ce să fie egal cu $E[XY \bmod p]$.

5. **Distributia geometrică**. James Bond este închis într-o celulă dar are trei posibilităti de evadare: evacuarea de pe sistemul de aer conditionat, o conductă de canalizare si usa (care nu este încuiată). În cazul încercării de evadare prin evacuarea aerului conditionat îl asteaptă o deplasare de două ore după care el cade printr-o trapă în cap, spre amuzamentul celor care-l tin închis. Utilizarea canalizării are un rezultat identic dar necesită cinci ore pentru a o traversa. Fiecare cădere produce amnezie temporară si el este returnat în celulă imediat după fiecare cădere. Se presupune că de fiecare dată Bond alege imediat una din cele trei iesiri cu probabilitatea de 1/3. Cât timp este necesar în medie pentru ca James Bond să realizeze că usa este

deschisă si să evadeze? *Indicatie*: Dacă recurgeti la calcule complicate, ati ales calea de rezolvare gresită.

Solutie: Punem X_i, variabilă aleatoare egală cu timpul cheltuit cu evadarea în încercarea i sau egală cu zero dacă evadarea este reusită în încercarea anterioară $i - 1$.

Bond esuează în încercările lui anterioare în număr de $i - 1$ cu probabilitatea $(2/3)^i$. Timpul de deplasare mediu la încercarea a i-a în conditiile eseculuí în cele $i - 1$ etape anterioare este $(1/3)(5 + 2 + 0) = 7/3$.

Astfel, $E[X_i] = (7/3)\ (2/3)^i$. Timpul total mediu consumat cu evadările nereusite este

$$\sum_{i=1}^{\infty} E[X_i] = \sum_{i=1}^{\infty} \frac{7}{3}\left(\frac{2}{3}\right)^i = \frac{7}{3}\frac{2}{3}\frac{1}{1-(2/3)} = \frac{14}{3}$$

1. **Utilizarea de unităti standard**. Fie Z o variabilă aleatoare care are o distributie normală cu media 0 si dispersia 1. Fiind dat un număr real z, există tabele care permit calculul $\Pr[Z \geq z]$ ca functie de z. De pildă:
   - "Coada" din dreapta: $\Pr[Z \geq 1] \approx 0{,}1587$. $\Pr[Z \geq 2] \approx 0{,}0228$. Coada din dreapta este aria de sub curba normală reprezentată de toate valorile mai mari sau egale cu z.
   - "Coada" din stânga: $\Pr[Z \leq -1] \approx 0{,}1587$. Coada din stânga este aria similară asociată valorilor mai mici sau egale cu z.

   Literatura si Internetul sunt între altele depozite de tabele si programe de evaluare a probabilitătilor de genul celor din exemplele de mai sus, numite uneori si z-scor.

(a) Fie Z o variabilă aleatoare distribuită normal cu media 0 si dispersia 1. Utilizati un tabel sau o rutină de calcul pentru a aproxima valoarea $\Pr[Z \geq 3]$.

Solutie: $\Pr[Z \geq 3] \approx 0{,}0013$.

(b) Se presupune că X este o variabilă aleatoare distribuită normal cu media 100 si abaterea medie pătratică (deviatia standard) 10. Calculati $\Pr[X \geq 125]$. Pentru utilizarea resurselor indicate mai sus, X trebuie mai întâi normalizată. Normalizarea se face cu formula $X_{\text{norm}} = (X - \mu)/\sigma$. X_{norm} este o variabilă aleatoare distribuită normal cu media 0 si dispersia 1. Asupra ei se pot face calculele necesare după care se revine la X cu relatia $X = \mu + \sigma X_{\text{norm}}$, inversa formulei de normalizare.

Solutie: Evenimentul $X \geq 125$ este echivalent cu $X_{\text{norm}} = (X - \mu)/\sigma \geq (125 - 100)/10 = 2{,}5$.

Asadar $\Pr[X \geq 125] = \Pr[X_{\text{norm}} \geq 2.5] \approx 0{,}0062$.

(c) Urmează o altă aplicatie a valorilor z-scor. Fie B numărul de succese într-o secventă de n experiente binomiale, fiecare cu probabilitatea de succes p. Am văzut că $E[B] = np$ si $Var[B] = np(1 - p)$. Pentru n mare, distributia binomială a lui B aproximează (este aproximată de) distributia normală cu aceeasi medie si aceeasi dispersie. Normalizarea lui B conduce la

$$B_{\text{norm}} = \frac{B - np}{\sqrt{np(1 - p)}}$$

si o seamă de rationamente se pot face prin intermediul valorilor z-scor asociate lui B_{norm}.

Găsiti o valoare a lui k pentru care, dacă o monedă este aruncată de 10.000 de ori, probabilitatea a k sau mai multe steme este aproximativ 0,2.

Solutie: Din tabele se obtine $\Pr[B_{\text{norm}} \geq 0{,}84] \approx 0{,}2$. Evenimentul $B_{\text{norm}} \geq 0{,}84$ este echivalent cu $B \geq np + 0{,}84\sqrt{np(1 - p)} = 5000 + 0{,}84\sqrt{2500} = 5042$, în

care s-a pus $n = 10000$ si $p = 1/2$. Asadar, probabilitatea de a obtine cel putin 5042 steme este de circa 0,2.

2. **Numerabile sau nenumerabile**? Determinati dacă multimile următoare sunt numerabile sau nenumerabile. Pentru fiecare multime numerabilă indicati o aplicatie biunivocă între multimea numerelor naturale si multimea respectivă, sau o enumerare a ei. Pentru fiecare multime nenumerabilă explicati de ce este nenumerabilă.

(a) Multimea sirurilor binare care sunt palindromice (care se pot scrie prin concatenarea a unui sir t cu t citit în ordine inversă).

Solutie: Numerabilă. Se poate crea o enumerare prin scrierea mai întâi a reprezentării binare a lui $n + 1$, stergând pe 1 initial si apoi atasând sirul citit în ordine inversă pentru a forma un palindrom. Aplicatia de la N la sirurile binare palindromice aplică pe 0, 1, 2, 3, … în ε, 00, 11, 0000, 0110, 1001, 1111, … cu ε sirul vid.

(b) Multimea numerelor reale din intervalul [0, 1] pentru care reprezentarea zecimală contine un singur 1 si toate celelalte cifre sunt zero.

Solutie: Numerabilă. Simplu, se aplică n pe 10^{-n}.

(c) Multimea numerelor reale din intervalul [0, 1] pentru care reprezentarea zecimală contine numai cifrele 0 si 1 (în orice ordine).

Solutie: Nenumerabilă. Fie S multimea descrisă în enunt. Dacă ar exista o bijectie $f: N \to S$ atunci am putea găsi un element $x \in S$ rămas înafara aplicatiei f. Folosim asa-numita diagonalizare pentru a găsi pe x. Simplu, fie zecimala a i-a a lui x 0 dacă digitul al i-lea din $f(i)$ este 1 si viceversa. Atunci $x \neq f(i)$ pentru orice $i \in N$ deoarece x diferă de orice $f(i)$ prin cel putin un digit zecimal.

(d) Multimea de arbori binari finiti cu rădăcină, în care arborii se disting numai prin formă (adică valorile din noduri sunt ignorate).

Solutie: Numerabilă. (Această solutie presupune că fiecare nod are 0 sau 2 copii, descendenti imediati). Se poate construi o enumerare prin listarea arborilor de adâncime n cu n de la 0 la infinit. Mai întâi listăm toti arborii de adâncime zero. Numai un astfel de arbore există, cel care constă numai din rădăcină. Apoi punem în listă arborii de adâncime 1. Din nou este vorba de un singur arbore, cel cu o rădăcină si doi descendenti imediati. Pentru a construi toti arborii de adâncime n, îi împărtim în următoarele categorii:

- Rădăcina, subarborele din stânga de adâncime $n - 1$, subarborele din dreapta de adâncime între 0 si $n - 2$
- Rădăcina, subarborele din stânga de adâncime între 0 si $n - 2$, subarborele din dreapta de adâncime $n - 1$
- Rădăcina, ambii subarbori de adâncime $n - 1$

Arborii de adâncime n sunt totdeauna în număr finit. Asadar, multimea arborilor binari este o reuniune numerabilă de multimi finite, este deci numerabilă.

3. **Multimea lui Cantor**. Multimea lui Cantor este un obiect straniu. Multimea este definită iterativ după cum urmează. Pornirea o constituie intervalul de numere reale [0, 1], inclusiv extremele. La prima iteratie se sterge treimea mijlocie (1/2, 2/3). Rămâne numai reuniunea celor două segmente-treimi [0, 1/3]∪ [2/3, 1]. În iteratia a doua se sterg treimile mijlocii ale acestor două segmente. Se continuă. În iteratia a *n*-a se elimină treimea mijlocie din fiecare din cele 2^{n-1} segmente rezultate din iteratia precedentă. Se notează cu *S* multimea de puncte care rămân după un număr nesfârsit de iteratii, adică $x \in S$ dacă *x* nu este sters în nici o iteratie.
   Se poate arăta că multimea lui Cantor este într-un sens foarte restrânsă. Dacă se ia un număr real *X* din intervalul [0, 1], la întâmplare dar uniform, atunci $\Pr[X \in S] = 0$. Cu toate acestea, multimea lui Cantor este într-un alt sens foarte cuprinzătoare.
   Arătati că multimea lui Cantor este nenumerabilă deci infinită.
   Indicatie: Sunt două căi consacrate de a arăta că o multime este nenumerabilă: găsirea unei bijectii cu o altă multime nenumerabilă sau utilizarea diagonalizării.
   Poate fi utilă o definitie alternativă a multimii lui Cantor: *S* este multimea numerelor din intervalul real [0, 1] care pot fi reprezentate în baza 3 numai prin cifrele 0 si 2 (niciodată 1). (Există o ambiguitate în reprezentarea ternară: 1/3 poate fi reprezentat si ca 0,100000… dar si ca 0,022222… Pentru această definitie ambiguitatea se elimină prin utilizarea totdeauna a reprezentărilor care se încheie în 022222… si nu în 100000… atunci când o alegere este necesară).

Solutie: Mai întâi, argumentăm că cele două definitii ale multimii lui Cantor sunt echivalente. Intervalul initial contine toate elementele dintre 0 si 1 fiecare putând fi reprezentat ca un sir ternar care începe cu 0,0, 0,1 sau 0,2. Există un număr egal de elemente care au fiecare din aceste începuturi deoarece subsirurile care le urmează pot fi la fel în fiecare caz. Totodată, toate numerele care încep cu 0,1 sunt mai mari decât cele care încep cu 0,0 si mai mici decât cele care încep cu 0,2 (ignorând punctele de capăt). Astfel, treimea mijlocie a elementelor sunt cele care încep cu 0,1 astfel că se elimină numai acele rezultate care încep cu 0,0 si 0,2.
Repetăm acest rationament în a doua iteratie pentru a elimina toate sirurile care încep cu 0,01 si 0,21 din treimea de eliminat toate acele numere care încep cu 0,001, 0,021, 0,201 si 0,221 si tot asa *ad infinitum*. Ceea ce rămâne sunt acele siruri care contin numai cifrele 0 si 2.
Acum putem executa diagonalizarea pentru a arăta că multimea lui Cantor este nenumerabilă. Presupunem că avem o bijectie de la *S* la *N*, o listă ordonată de elemente din *S*, (s_1, s_2, …). Putem construi un alt număr *s'* astfel încât în pozitia *i*-a de după virgulă *s'* să aibă un 0 dacă s_i are un 2 în acea pozitie, *s'* să aibă acolo un 2 dacă s_i are un 0 în pozitia *i*. Astfel, *s'* diferă de oricare din numerele s_i din *S*. Dar se vede că *s'* contine numai cifrele 0 si 2, niciodată 1, prin urmare apartine multimii *S*. Această contradictie arată că bijectia presupusă nu există si nici vreo alta. Asadar, *S* este nenumerabilă.

O solutie alternativă constă în a da o bijectie între S si o multime nenumerabilă. Reamintim că multimea de numere reale $R_{0,1}$ = [0, 1] este nenumerabilă. O bijectie simplă între S si $R_{0,1}$ se obtine prin înlocuirea fiecărui 2 care apare într-un element din S cu un 1, pentru a obtine un număr real din intervalul [0, 1] care este scris în binar. Este usor de văzut că aceasta este o bijectie. Asadar, S este nenumerabilă.

4. **Prea multe functii**? Fie S multimea de functii de la multimea N la multimea {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}. Arătati că S este nenumerabilă.
   Indicatie: Stabiliti o corespondentă biunivocă între multimea numerelor reale din intervalul [0, 1] si o submultime din S.

Solutia 1: Se foloseste diagonalizarea. Se admite că există o bijectie ϕ între N si S. Atunci se poate crea o functie $f \in S$ care nu este acoperită de bijectia ϕ. Simplu, se ia $f(n) = \phi_n(n) + 1 \bmod 10$ pentru orice $n \in N$ astfel încât $f(n) \neq \phi_n(n)$ si $f \neq \phi_n$ pentru orice $n \in N$. (De retinut că ϕ_n este o *functie*.). Asadar nu există nici o bijectie ϕ, prin urmare S este nenumerabilă.

Solutia 2: Se consideră bijectia care aplică fiecare număr real pe o functie din S după cum se explică imediat. Se ia numărul $0,d_1d_2d_3\ldots$ Functia corespunzătoare f are $f(k) = d_k$ pentru orice k. Pentru orice functii distincte f_1 si f_2 din S, trebuie să existe un $k \in N$ pentru care $f_1(k) \neq f_2(k)$ astfel că f_1 si f_2 sunt aplicate în numere reale distincte. Invers, două numere reale diferite pentru care digitii din pozitia k nu sunt identici se aplică pe functii distincte. Deoarece multimea de numere reale din [0, 1] este nenumerabilă, la fel este si multimea S.

5. **QuickSelect**. O cale de a găsi cea de a k cea mai mică valoare dintr-o multime S de n întregi ($n > k$) este a sorta S într-un masiv de date; cea de a k cea mai mică valoare a masivului va fi valoarea dorită. Cum s-a văzut, pentru asta sunt necesare $n\log n$ comparatii.
   Cu un algoritm numit QuickSelect se poate găsi cea de a k cea mai mică valoare într-o multime de n întregi distincti într-un timp liniar. Iată algoritmul.

   QuickSelect(S, k)
   alege $p \in S$ la întâmplare;
   $lowVals = \{x \in S: x < p\}$;
   $highVals = \{x \in S: x > p\}$;
   if $k = size(lowVals) + 1$ then returnează p
   else if $k \leq size(lowVals)$ then returnează *QuickSelect*($lowVals$, k)
   else returnează *QuickSelect*($highVals$, $k - size(lowVals) - 1$);

   Cazul cel mai bun, când prima valoare aleasă pentru p este acel element k, necesită $n - 1$ comparatii. Cazul cel mai rău, când p este totdeauna ales ca elementul maxim din S si $k = 1$, necesită $n(n - 1)/2$ comparatii. Numărul

mediu de comparatii, totusi – să-l numim $T(n)$ – este mai mic decât $4n$. Dovediti acest adevăr.

Solutie: Fie $T(n, k)$ numărul mediu de comparatii necesare pentru a găsi al k cel mai mic întreg din n întregi. $T(n, k)$ se poate exprima în functie de alte valori $T(n', k')$. Pentru orice alegere a lui p, sunt necesare $n - 1$ comparatii pentru a separa pe S în *lowVals* si *highVals*. Fie s dimensiunea lui *lowVals*. Dacă $s > k$, atunci trebuie rulat QuickSelect pentru a afla elementul k cel mai mic dintr-o grupă de s întregi. Vor fi necesare în medie $T(s, k)$ comparatii pentru a afla întregul dorit. Dacă $s + 1 > k$, atunci trebuie rulat QuickSelect pentru a găsi elementul al $(k - s - 1)$ din cele $n - s - 1$, ceea ce necesită în medie $T(n - s - 1, k - s - 1)$ comparatii suplimentare. În final, dacă $s + 1 = k$, treaba este încheiată, nu mai este nevoie de alte comparatii. Fiecare din aceste cazuri are probabilitatea de aparitie $1/n$. Astfel $T(n, k)$ se poate exprima ca o sumă

$$T(n,k) = n - 1 + \frac{1}{n}\left(\sum_{s=0}^{k-2} T(n-s-1, k-s-1) + 0 + \sum_{s=k}^{n-1} T(s,k)\right)$$

Se poate arăta că $T(n, k) < 4n$ prin inductie după n, indiferent de valoarea lui k. În cazurile de bază $n = 0$ si $n = 1$, nu sunt necesare comparatii. Pentru pasul inductiv, utilizăm faptul că $T(s, k) < 4s$ pentru orice $s < n$.

$$T(n,k) < n - 1 + \frac{1}{n}\left(\sum_{s=0}^{k-2} 4(n-s-1) + \sum_{s=k}^{n-1} 4s\right) =$$

$$= n - 1 + \frac{1}{n}\left(\frac{k-1}{2}[4(n-1) + 4(n-k-1)] + \frac{n-k}{2}[4k + 4(n-1)]\right) =$$

$$= n - 1 + [2(k-1)(2n-k) + 2(n-k)(k+n-1)]/n =$$

$$= n - 1 + 2[n^2 - 3n - 2k^2 + 2k + 2kn]/n$$

Cantitatea $-2k^2 + 2k + 2kn$ are o valoare maximă de $(n + 1)^2/2$ pentru $k = (n + 1)/2$. Valoarea aceasta se obtine prin anularea derivatei în raport cu k. Se obtine o limitare superioară a lui $T(n, k)$:

$$n - 1 + 2n - 6 + (n + 1)^2/n = 3n - 7 + n + 2 + 1/n = 4n - 5 + 1/n < 4n$$

Asadar, $T(n) < 4n$ pentru orice n.