
Algoritmi și Structuri de Date I. Seminar 1: Rezolvarea algoritmică a problemelor

- De la ce pornim? Ce se dă și ce se cere?
- Putem abstractiza problema? Este problema similară unor probleme pe care le-am rezolvat anterior?
- Este vreo diferență între modul în care rezolvăm problemele în matematică și abordarea din informatică?

Problema 1 (*Examen admitere la Informatică.*) La examenul de bacalaureat am obținut media generală 7.50, nota la proba de Matematică fiind 7.25 iar la Informatică 8.75. Care este nota minimă pe care ar trebui să o obțin la proba scrisă a examenului de admitere pentru ca media de admitere să fie cel puțin 5?

Extras din Metodologia de admitere la specializările de Informatică (2018-2019):

La domeniul Informatică media de admitere se calculează ca $N = 1/2 * (N3 + \max\{N1, N2\})$, unde $N1$ este media generală la examenul de bacalaureat, $N2$ este nota la proba de matematică sau informatică de la examenul de bacalaureat (lipsa notei se echivalează cu nota 4) iar $N3$ este nota obținută la proba scrisă de admitere la matematică sau informatică.

Indicație. Datele de intrare sunt: $N1$ (media generală la examenul de bacalaureat), $N2m$ (nota la proba de Matematică de la bacalaureat), $N2i$ (nota la proba de Informatică de la bacalaureat), $Nmin$ (nota minimă de promovare a examenului de admitere). Se cere să se determine cea mai mică valoare a lui $N3$ care asigură că N este mai mare decât $Nmin$. Din punct de vedere matematic problema este echivalentă cu rezolvarea inecuației: $(N3 + \max\{N1, N2m, N2i\})/2 \geq Nmin$.

Prin rezolvarea inecuației se obține $N3 \geq 2Nmin - \max\{N1, N2m, N2i\}$. Din punct de vedere matematic, problema este rezolvată. Din punct de vedere informatic este necesară descrierea secvenței de prelucrări care permite calculul marginii inferioare determinate matematic, ceea ce presupune următoarea secvență de prelucrări:

- determinarea maximului dintre trei valori;
- calculul produsului dintre 2 și $Nmin$;
- calculul diferenței.

Cum ar putea fi rezolvată problema fără să recurgem la rezolvarea inecuației? (presupunem că nu știm deloc matematică și nu am auzit de inecuații).

Problema 2 (*Înmulțirea à la russe.*) Se consideră următoarea metodă de înmulțire (numită înmulțirea "à la russe") a două numere naturale nenule x și y : "Se scrie x alături de y (pe aceeași linie). Se împarte x la 2 și câtul împărțirii se scrie sub x (restul se ignoră deocamdată). Se înmulțește y cu 2 iar produsul se scrie sub y . Procedura continuă construindu-se astfel două coloane de numere. Calculele se opresc în momentul în care pe prima coloană se obține valoarea 1. Se adună toate valorile de pe coloana a doua care corespund unor valori impare aflate pe prima coloană."

Exemplu. Fie $x = 13$ și $y = 25$. Succesiunea de rezultate obținute prin aplicarea operațiilor de mai sus este:

| x | y | rest | factor multiplicare y |
|-------|-----|------|-------------------------|
| 13 | 25 | 1 | 2^0 |
| 6 | 50 | 0 | 2^1 |
| 3 | 100 | 1 | 2^2 |
| 1 | 200 | 1 | 2^3 |
| <hr/> | | | |
| 325 | | | |

a) Prelucrarea descrisă prin metoda de mai sus se termină întotdeauna după un număr finit de pași? Ce se întâmplă în cazul în care una dintre valori este egală cu 0?

b) Metoda de mai sus conduce întotdeauna la produsul celor două numere (cu alte cuvinte este o metodă corectă de înmulțire)?

c) Câte operații de înmulțire cu 2 sunt necesare? Depinde acest număr de ordinea factorilor?

Indicație.

a) Ca urmare a împărțirilor succesive la 2 valorile de pe prima coloană descresc până se ajunge la un cât egal cu 1 (în ipoteza că x este nenul). Prin urmare prelucrarea este finită. Dacă x este egal cu 0 metoda nu poate fi aplicată ca atare, însă dacă y este 0 ea conduce la un rezultat corect.

b) Corectitudinea prelucrării derivă din faptul că metoda realizează de fapt conversia în baza 2 a primului număr (cifrele reprezentării în baza doi sunt resturile obținute prin împărțirile succesive la 2) iar produsul se obține prin înmulțirea succesivă a celui de al doilea număr cu puteri ale lui 2 și prin însumarea acelor produse care corespund unor cifre nenule în reprezentarea binară a primului număr.

c) Numărul înmulțirilor cu 2 este cu 1 mai mic decât numărul de cifre ale reprezentării binare a lui x , adică $\lfloor \log_2 x \rfloor$. Evident numărul înmulțirilor depinde de ordinea factorilor fiind mai mic dacă împărțirile la 2 se efectuează asupra celui mai mic număr.

Link: <http://mathforum.org/dr.math/faq/faq.peasant.html>

Problema 3 (*Determinarea părții întregi.*) Propuneți o metodă bazată pe operații de adunare, scădere și comparare pentru determinarea părții întregi a unui număr real. Partea întreagă (numită uneori parte întreagă inferioară și notată pentru un număr real x cu $\lfloor x \rfloor$) a unui număr real este definită ca fiind cel mai mare număr întreg mai mic decât numărul real respectiv. De exemplu, partea întreagă a lui 2.4 este 2, iar pentru -2.4 este -3 .

Indicație. Se va ține cont de semnul numărului. În cazul în care este pozitiv se scade succesiv valoarea 1 până când se ajunge la o valoare mai mică strict decât 1. Numărul scăderilor efectuate indică valoarea părții întregi. Pentru numere negative se adună succesiv 1 până se obține o valoare mai mare sau egală cu 0. Opusul numărului de adunări reprezintă valoarea părții întregi.

Exemplu.

| $x > 0$ | | $x < 0$ | |
|---------|--------|---------|--------|
| x | contor | x | contor |
| 3.25 | 0 | -3.25 | 0 |
| 2.25 | 1 | -2.25 | -1 |
| 1.25 | 2 | -1.25 | -2 |
| 0.25 | 3 | -0.25 | -3 |
| | | 0.75 | -4 |

Problema 4 (*Problema identificării monedei mai ușoare.*) Se consideră un set de n monede identice, cu excepția uneia care are greutatea mai mică decât a celorlalte. Folosind o balanță simplă (care permite doar compararea greutății monedelor plasate pe cele două talere) să se identifice moneda cu greutatea mai mică folosind cât mai puține comparații.

Indicație. O primă variantă este aceea prin care se selectează la întâmplare două monede și se pun pe balanță. Dacă una dintre ele are greutatea mai mică atunci este chiar moneda căutată. Dacă ambele au aceeași greutate se păstrează una dintre ele pe un taler al balanței iar pe celălalt taler se pun succesiv monedele rămase. Prelucrarea se oprește în momentul în care se găsește o monedă cu greutatea mai mică. Numărul de comparații este cel mult $n - 1$.

O altă variantă este cea în care se împarte setul de monede în două subseturi (fiecare va avea $n/2$ elemente, dacă n e par, respectiv $(n-1)/2$ dacă n este impar; în cazul în care n este impar se pune o monedă deoparte) care se pun pe cele două talere ale balanței. Dacă subseturile au aceeași greutate (ceea ce se poate întâmpla doar dacă n este impar) atunci moneda pusă deoparte este cea căutată. În caz contrar se aplică același procedeu subsetului de greutate mai mică și se continuă până se ajunge la un set de două sau trei monede. În acest moment este suficient să se mai efectueze o singură cântărire. Numărul de cântăriri este cel mult $\lfloor \log_2 n \rfloor$. Este adevărat acest rezultat și pentru cazul în care se știe doar că una dintre monede este diferită dar nu se știe dacă este mai ușoară sau mai grea decât celelalte?

Problema 5 (*Problema clătitorilor.*) La un restaurant bucătarul a pregătit clătite (americane) pe care le-a așezat pe un platou sub forma unei stive. Din păcate nu toate clătitele au același diametru astfel că stiva nu arată estetic. Chelnerul ia platoul și având la dispoziție o spatulă reușește să aranjeze cu o singură mână clătitele astfel încât să fie în ordinea descrescătoare a diametrelor (cea mai mare clătită pe platou iar cea mai mică în vârful stivei). Cum a procedat? Descrieți problema într-o manieră abstractă și propuneți un algoritm de rezolvare.

Indicație. Rearanjarea se face efectuând doar mișcări de răsturnare a unui "set" de clătite dintre cele aflate în partea de sus a stivei. Problema este identică cu cea a ordonării descrescătoare a unui șir de valori prin inversarea ordinii elementelor unor subsecvențe de la sfârșitul șirului.

Exemplu. Pentru a ordona descrescător șirul (5, 3, 4, 1, 6, 2) se aplică următoarea secvență de prelucrări în care subsecvența care se "răstoarnă" este încadrată:

```

5 3 4 1 6 2
5 3 4 1 2 6
6 2 1 4 3 5
6 5 3 4 1 2
6 5 3 2 1 4
6 5 4 1 2 3
6 5 4 3 2 1

```

Metoda constă în determinarea valorii maxime din secvență și inversarea ordinii subsecvenței care începe cu ea pentru a fi adusă pe ultima poziție în șir. După aceea se răstoarnă întregul șir, astfel că valoarea maximă ajunge pe prima poziție. Se aplică aceeași metodă pentru subsecvența care începe cu al doilea element și se continuă până se ajunge la o subsecvență constituită dintr-un singur element.

Problema 6 (*Problema partiționării echilibrate.*) Considerăm un set de programe pentru care se cunosc estimări ale timpilor de execuție și care trebuie executate utilizând două calculatoare similare. Propuneți o modalitate de distribuire a execuției programelor pe cele două calculatoare astfel încât încărcarea acestora să fie cât mai echilibrată.

Indicație. Problema poate fi reformulată (abstractizată) astfel: se consideră o mulțime de numere pozitive, $A = \{a_1, a_2, \dots, a_n\}$, $n > 1$ și se cere să se determine două submulțimi disjuncte B și C astfel încât $B \cup C = A$ și $|\sum_{a \in B} a - \sum_{a \in C} a|$ este minimă. Cea mai simplă metodă este cea a "forței brute" prin care se generează toate perechile de submulțimi (B, C) și se alege cea care minimizează diferența specificată. Numărul de partiții distincte (fără a ține cont de ordinea calculatoarelor) este $2^{n-1} - 1$. Pentru n mare, numărul de partiții ce trebuie testate devine mare (pentru $n = 10$ este 511 iar pentru $n = 100$ este de ordinul 10^{29}). E evident că în astfel de situații metoda forței brute nu este eficientă, astfel că trebuie căutate metode mai puțin costisitoare (astfel de metode vor fi discutate în seminariile viitoare).

Problema 7 (*Problema acoperirii.*) Se consideră o mulțime cu n elemente (de exemplu $U = \{1, 2, \dots, n\}$) și un set S de submulțimi ale acesteia. Se pune problema determinării unei acoperiri minimale a mulțimii U folosind submulțimi din setul S (o acoperire minimală este un subset $A \subset S$ cu proprietatea că $\cup_{s \in A} s = U$ iar numărul de elemente din A este minim). De exemplu pentru $n = 5$ se poate considera setul de submulțimi $S = \{S_1, S_2, S_3, S_4\}$ cu $S_1 = \{1, 3, 5\}$, $S_2 = \{1, 4\}$, $S_3 = \{2, 4\}$, $S_4 = \{2, 5\}$. În acest caz acoperirea minimală se obține selectând prima dată submulțimea cu cele mai multe elemente (S_1), eliminând din U elementele submulțimii alese (U devine $\{2, 4\}$) și continuând prin selectarea unei submulțimi care conține elemente din U (în acest exemplu ar fi S_3). Garantează strategia de a alege la fiecare etapă cea mai numeroasă mulțime care acoperă elementele rămase obținerea unei soluții optime? Dacă nu, propuneți un contraexemplu.

Indicație. Strategia de a alege tot timpul cea mai mare submulțime nu garantează obținerea unei soluții optime. Pentru $n = 6$ un contraexemplu este $S_1 = \{1, 3, 5\}$, $S_2 = \{2\}$, $S_3 = \{4\}$, $S_4 = \{6\}$, $S_5 = \{1, 2\}$, $S_6 = \{3, 4\}$, $S_7 = \{5, 6\}$.

Probleme suplimentare.

1. Se pune problema determinării valorii în baza 10 a unui număr natural pornind de la șirul cifrelor sale binare. Propuneți o metodă bazată doar pe operații elementare (adunare, scădere, înmulțire - ridicarea la putere nu este considerată operație elementară). Se urmărește ca numărul operațiilor implicate să fie cât mai mic.
2. Se consideră un număr constituit din 10 cifre distincte, diferit de 9876543210. Să se determine numărul care îl succede în șirul crescător al tuturor numerelor naturale ce conțin 10 cifre distincte.
3. Se consideră un șir de valori nenule (pozitive și negative). Să se transforme șirul astfel încât toate valorile negative să le preceadă pe cele pozitive iar numărul operațiilor efectuate să fie cât mai mic.
4. Se consideră un șir cu $n - 1$ valori distincte din mulțimea $\{1, 2, \dots, n\}$. Să se determine, folosind un număr cât mai mic de comparații, valoarea care lipsește din șir.
5. Se consideră următorul joc cu doi jucători bazat pe două stive de monede: una conținând m monede și cealaltă conținând n monede. La fiecare etapă jucătorul care este la rând poate ridica o monedă dintr-una dintre stive sau câte o monedă din ambele stive. Câștigă jucătorul care ridică ultima/ultimele monede. Propuneți un algoritm care pentru o pereche de valori (n, m) decide dacă pentru jucătorul care face prima mutare există o strategie câștigătoare.