

**Tema 9**  
**19 decembrie 2016**

*Problemă obligatorie*

**Termen de predare :** Laboratorul din săptămâna 13 (9 ianuarie 2017)

(3 p) **1.** Sa se implementeze o coadă cu priorități folosindu-se un heap (Cormen, capitolul 7.5). Elementele cozii vor avea doua câmpuri: prioritate și cheie. Vor exista următoarele operații:

- `insert(q, x)` care inserează nodul `x` în coada `q`;
- `maximum(q)` care întoarce elementul de prioritate maximă din coada `q`;
- `extract_max(q)` care întoarce elementul de prioritate maximă din `q`, eliminându-l din coadă.

*Probleme suplimentare*

**Termen de predare :** Laboratorul din săptămâna 13 (9 ianuarie 2017)

(2 p) **2.** Să se implementeze algoritmul *Shell-Sort* folosind ca tablou de incremenți unul dintre șirurile propuse în materialul ajutător alăturat.

(4 p) **3. Roata**

---

Una dintre atracțiile celebrului parc de distracții Prater din Viena este Marea Roată Vienneză. Din ea se poate admira priveliștea întregii Viene.

Roata are  $n$  cabine, numerotate de la 1 la  $n$  în sens orar și dispuse simetric pe circumferința roții. Îmbarcarea clienților se face în cabina în care roata este tangentă cu solul, iar rotirea începe cu cabina 1 aflată în poziția de îmbarcare și se face în sens antiorar. Un client plătește pentru o rotire 1 EUR și poate cumpăra un număr oarecare de rotiri.

Cei  $p$  clienți care doresc utilizarea roții trebuie să respecte următoarea procedură: clientul cu numărul de ordine  $i$  își cumpără un bilet pe care sunt înscrise numărul său de ordine și numărul de rotiri  $c_i$ ,  $1 \leq i \leq p$ , apoi se așează la rând. Când în poziția de îmbarcare este o cabină liberă sau se eliberează o cabină, roata se oprește și urcă următorul client. Un client coboară după ce se efectuează numărul de rotiri înscris pe bilet.

### Cerință

Să se scrie un program care, cunoscând numărul  $n$  de cabine al roții, numărul  $p$  de clienți, precum și numărul de rotații cumpărate de fiecare client,  $c_i$ ,  $1 \leq i \leq p$ , să calculeze:

- suma totală încasată de administratorul roții de la clienți;
- ordinea în care coboară clienții din roată;
- numărul cabinei din care coboară ultimul client.

### Date de intrare

Fișierul de intrare `roata.in` conține pe primul rând numărul natural  $n$ , pe al doilea rând numărul natural  $p$  iar pe al treilea rând numerele naturale  $c_i$ ,  $1 \leq i \leq p$ , separate printr-un spațiu, cu semnificațiile de mai sus.

### Date de ieșire

Fișierul de ieșire `roata.out` va conține pe prima linie suma totală încasată, pe a doua linie numerele de ordine ale clienților, în ordinea coborării, separate printr-un spațiu, iar pe a treia linie numărul cabinei din care va coborî ultimul client.

### Restricții

- $2 \leq n \leq 360$
- $1 \leq p \leq 100\,000$
- $1 \leq c_i \leq 100\,000$
- pentru rezolvarea primei cerințe se acordă 20% din punctaj, iar pentru celelalte două cerințe se acordă câte 40% din punctaj fiecare.

### Exemplu

roata.in	roata.out	Explicație
4 7 6 4 1 5 2 8 3	29 3 5 2 4 1 7 6 3	Roata are $n = 4$ cabine și numărul de clienți este $p = 7$ . Primul client cumpără 6 rotații, al doilea 4 rotații, ..., iar al șaptelea client cumpără 3 rotații. Suma totală încasată este de 29 EUR. După ce primii 4 clienți se urcă în roată și se efectuează o rotație completă, primul care coboară este clientul al 3-lea și imediat se urcă clientul al 5-lea. După încă 2 rotații, clientul al 5-lea coboară și se urcă clientul al 6-lea. După încă o rotație coboară clientul al 2-lea și se urcă al 7-lea client. Ultimii 4 clienți coboară în ordinea 4, 1, 7, 6. Cabina din care coboară ultimul client este cabina cu numărul 3

(3 p) **4. Bitone**

O secvență de numere întregi se numește **bitonă** dacă este crescătoare la început, iar apoi descrescătoare. Mai precis, o secvență  $a_1, a_2, \dots, a_n$  este bitonă dacă:

- este o secvență nedescrescătoare:  $a_1 \leq a_2 \leq \dots \leq a_n$  **sau**
- este o secvență necrescătoare:  $a_1 \geq a_2 \geq \dots \geq a_n$  **sau**
- exista un indice  $i$  pentru care  $a_1 \leq a_2 \leq \dots \leq a_i \geq a_{i+1} \geq \dots \geq a_n$

*Cerință*

Data o secvență de numere întregi  $a_1, a_2, \dots, a_n$  și niște întrebări de forma  $(i, j)$  să se răspundă pentru fiecare întrebare dacă subsecvența  $a_i, a_{i+1}, \dots, a_j$  este bitonă.

*Date de intrare*

Fișierul de intrare bitone.in conține pe prima linie numărul de numere din secvență,  $n$ . Pe a doua linie conține cele  $n$  numere ale secvenței, separate de spații. Pe a treia linie se află numărul de întrebări  $q$ . Pe următoarele  $q$  linii se vor găsi câte două numere  $i, j$  separate prin spațiu, reprezentând întrebările la care se cere răspuns.

*Date de ieșire*

Fișierul de ieșire bitone.out va conține o singură linie cu  $q$  caractere 0 sau 1, fără spații între ele, caractere ce reprezintă răspunsurile la întrebări. Pentru fiecare întrebare veți răspunde 1 dacă subsecvența este bitonă, sau 0 în caz contrar.

*Restricții*

- $1 \leq n \leq 1.000.000$
- $-2.000.000.000 \leq a_i \leq 2.000.000.000$
- $1 \leq q \leq 1.000.000$
- $1 \leq i \leq j \leq n$

*Exemple*

bitone.in	bitone.out	Explicație
10 10 19 19 18 18 21 21 11 11 13 6 9 10 6 10 4 8 8 10 1 7 3 3	101101	subsecvențele (6, 10) și (1, 7) nu sînt bitone. Toate celelalte sînt.

15 10 11 13 13 6 8 8 8 4 4 5 9 0 2 2 10 2 10 9 13 1 3 7 14 4 7 1 9 3 10 4 11 13 13 9 9	0110000011	subsecvențele (9, 13) (1, 3) (13, 13) și (9, 9) sînt bitone. Toate celelalte nu.
--	------------	--

Cerc informatică Vianu

### (3 p) 5. La coadă

La BIG au băgat pui<sup>1</sup>. Instantaneu s-a format o coadă de N persoane, numerotate în ordine de la 1 la N. La coadă se pot întâmpla următoarele lucruri:

1. Servire: prima persoană de la coadă primește un pui și pleacă acasă.
2. Sosire: la coadă se mai așează o persoană. Noii veniți sunt numerotați în continuare:  $N + 1$ ,  $N + 2$  ș.a.m.d.
3. Îmbrâncire(x): persoana numărul x face rost de o relație și se îmbrânțește până pe prima poziție a cozii. Dacă persoana era deja prima, nu se schimbă nimic.

Se dă o listă de K operații. Să se spună care este configurația finală a cozii. Se garantează că în niciun moment lungimea cozii nu va depăși N (oamenii se descurajează dacă văd o coadă prea lungă și nu se mai așează). Se garantează că operațiile de servire și îmbrâncire nu se vor efectua pe o coadă goală.

#### *Date de intrare*

Fișierul de intrare lacoada.in conține pe prima linie numerele N și K. Pe următoarele K linii se vor găsi operațiile, numerotate ca mai sus, într-una din formele

1

2

3 x

Se garantează că x este numărul unei persoane din coadă.

#### *Date de ieșire*

În fișierul de ieșire lacoada.out se va tipări pe prima linie lungimea cozii la sfârșitul operațiilor. Pe a doua linie se vor tipări, în ordine, numerele persoanelor de la coadă, începând cu prima.

### Restricții

- $1 \leq N \leq 60.000$
- $1 \leq K \leq 1.000.000$

### Exemplu

lacoada.in	lacoada.out	Explicație
6 6	5	5 se îmbrânțește, coada devine 5 1 2 3 4 6
3 5	3 1 2 4 6	5 este servit, coada devine 1 2 3 4 6
1		3 se îmbrânțește, coada devine 3 1 2 4 6
3 3		7 sosește, coada devine 3 1 2 4 6 7
2		7 se îmbrânțește, coada devine 7 3 1 2 4 6
3 7		7 este servit, coada devine 3 1 2 4 6
1		

*Autor:* Cătălin Frâncu

## Probleme facultative

**Termen de predare :** Laboratorul din săptămâna 12 (7 ianuarie 2016)

(5 ps) 1. Spunem ca o tabla de sah de  $2^k \times 2^k$  patrate este defecta, daca unul din cele  $2^{2k}$  patrate lipseste. Problema va cere sa acoperiti o astfel de tabla cu tromino-uri (Figura 1), astfel incat oricare doua tromino-uri nu se suprapun, ele nu acopera patratul lipsa, dar acopera toate celelalte patrate. Sugestii de implementare:

(a) o acoperire a unei table  $m \times m$  se poate reprezenta printr-o matrice  $\text{Tabla}[m][m]$ , unde  $\text{Tabla}[i][j]$  indica numarul trominoului cu care este acoperit patratul  $(i; j)$ .

(b) Functia recursiva ce construiesc solutia poate fi de forma:  $\text{Acopera}(\text{rt}, \text{ct}, \text{rd}, \text{cd}, \text{latura})$ , unde :

- i.  $\text{rt}, \text{ct}$  reprezinta randul si coloana patratului din coltul stanga sus al portiunii patratice de tabla ce trebuie acoperita;
- ii.  $\text{rd}, \text{cd}$  reprezinta randul si coloana patratului lipsa;
- iii.  $\text{latura}$  reprezinta latura portiunii patratice de tabla ce trebuie acoperita.

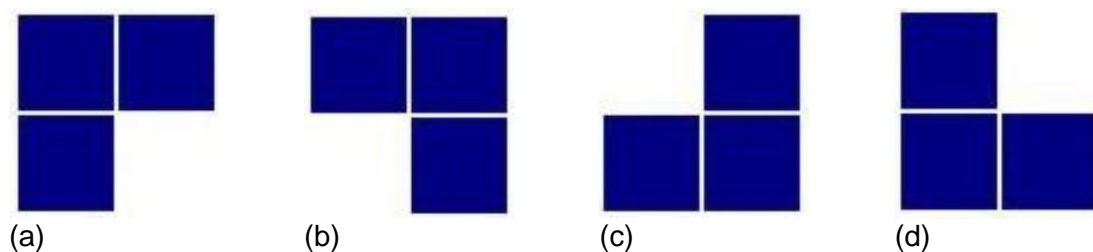


Figura 1. Tromino-uri

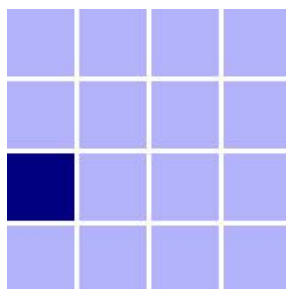


Figura 2. O tablă de șah defectă de dimensiuni  $2^2 \times 2^2$