

**Tema 8**  
**12 decembrie 2014**

*Problemă obligatorie*

**Termen de predare :** Laboratorul din săptămâna 13 (9 ianuarie 2015)

*1. Sortări*

(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 2015)

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

*2. Tehnici de programare*

(2 p) **3.** Dat un număr natural  $n$ , reprezentând restul pe care o persoană trebuie să îl primească după efectuarea unei plăți, să se spună care este numărul minim de bancnote utilizate pentru plata restului. Presupunem că există  $k$  tipuri de bancnote, cu valori  $b_1; b_2; \dots, b_k$ . Datele de intrare se citesc din fișierul de intrare `input.txt`. Iesirea se afișează pe ecran.

Exemplu:

$n = 1242; k = 4; b_1 = 90; b_2 = 25; b_3 = 6; b_4 = 3$

Restul poate fi plătit ca  $1242 = 10 \times b_1 + 12 \times b_2 + 5 \times b_3 + 4 \times b_4$ . Soluția nu este unică.

(2 p) **4.** Un hotel este foarte faimos pentru sala sa de conferințe. Acesta a primit  $n$  cereri de tipul  $[s; f)$  de a închiria sala de conferințe în intervalul de timp cuprins între  $s$  (inclusiv) și  $f$  (exclusiv). Pentru că fiecare închiriere aduce proprietarilor hotelului un venit fix, aceștia ar dori să onoreze cât mai multe cereri. Sala nu poate fi închiriată la

doua persoane in acelasi timp. Sa se spuna care este numarul maxim de cereri care pot fi onorate si care sunt acestea. Datele de intrare se citesc din fisierul `input.txt` astfel: pe primul rand se afla  $n$ , iar pe urmatoarele  $n$  randuri cate o pereche  $s\ f$ , reprezentand intervalul  $[s; f)$ .

*Exemplu:*

```
7
0 2
3 7
4 7
9 11
7 10
1 5
6 8
```

Numarul maxim de cereri este 3, iar cererile pot fi  $[0; 2)$ ;  $[3; 7)$ ;  $[7; 10)$ .

### *Probleme facultative*

**Termen de predare :** Laboratorul din săptămâna 13 (9 ianuarie 2015)

(10 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 `Tabla[m][m]`, unde `Tabla[i][j]` indica numarul trominoului cu care este acoperit patratul  $(i; j)$ .

(b) Functia recursiva ce construiesc solutia poate fi de forma: `Acopera(rt, ct, rd, cd, latura)`, unde :

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

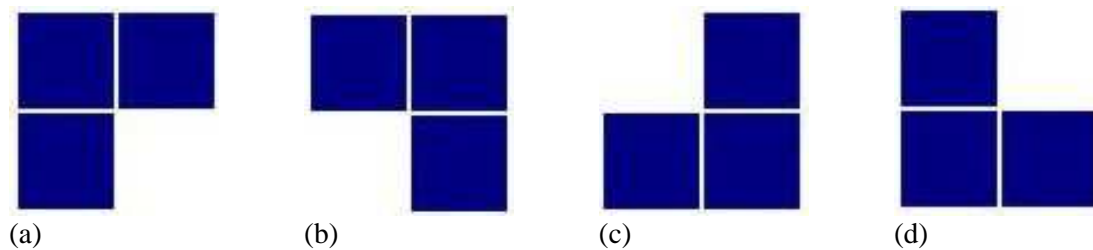


Figura 1. Tromino-uri

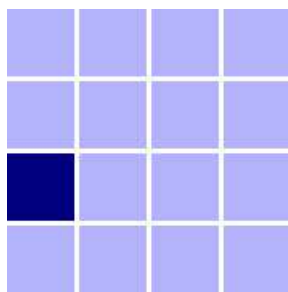


Figura 2. O tablă de șah defectă de dimensiuni  $2^2 \times 2^2$