

## NR 1

1. (4p) Se dau  $n$ ,  $m$  și  $k$ ,  $k \geq 1$ ,  $k \leq n+m$ , numere naturale și doi vectori  $a$  și  $b$  de lungime  $n$ , respectiv  $m$ , cu elementele ordonate crescător. Propuneți un algoritm cât mai eficient pentru a determina al  $k$ -lea element al vectorului obținut prin interclasarea celor doi vectori. Complexitate:  $O(\log(\max\{n,m\}))$  – abstracție făcând de citirea vectorului.

Spre exemplu, pentru fișierul *intrare.txt*

```
6, 4, 7
2 4 8 9 10 12
1 5 6 13
```

se va afișa pe ecran

```
9
```

2. (5p) Se consideră un dicționar de cuvinte  $d$  și un șir  $s$  fără spații. Fiecare cuvânt din dicționar are asociată o pondere. Să se obțină în câte moduri poate fi  $s$  partiționat în cuvinte din dicționar prin adăugarea de spații astfel încât suma ponderilor cuvintelor care formează partiția  $s$  să fie **maximă**.

Se citește din fișierul "intrare.txt" un număr natural  $N$ , urmat de șirul  $S$ , urmat din  $N$  linii, reprezentând cuvintele din dicționar și ponderile acestora. Se va afișa pe ecran costul partiționării optime și numărul modurilor în care se poate obține această împărțire. Dacă  $s$  nu poate fi partiționat se va afișa -1.

Complexitate  $O(n^2)$

intrare.txt	iesire
8 programaredinamica program 3 are 1 pro 2 gram 1 programare 2 dinamica 4 dinamic 2 a 2	8 4

program + are + dinamic + a -- cost 8  
pro + gram + are + dinamica -- cost 8  
program + are + dinamica -- cost 8  
pro + gram + are + dinamic + a -- cost 8  
programare + dinamica -- cost 6

### Observatii.

1. Se acordă **1p** din oficiu. Nota reprezintă 30% din nota finală
2. Se va preda un director/fișiere cu numele **grupa\_Nume\_Prenume\_Nr** (se pot pune în director proiectul sau doar fișiere sursă, se pot rezolva cerințele 1 și 2 în metode/clase/fișiere separate)
3. Se va scrie în comment o adresă de email pentru primirea rezultatelor

## NR 2

1. . (4p) Se dau  $n$ ,  $m$  și  $k$ ,  $k \geq 1$ ,  $k \leq n+m$ , numere naturale și doi vectori  $a$  și  $b$  de lungime  $n$ , respectiv  $m$ , cu elementele ordonate descrescător. Propuneți un algoritm cât mai eficient pentru a determina al  $k$ -lea, cel mai mic element al vectorului obținut prin interclasarea celor doi vectori.

Complexitate:  $O(\log(\max\{n,m\}))$  – abstracție făcând de citirea vectorului.

Spre exemplu, pentru fișierul *intrare.txt*

```
4, 7, 3
17 15 4 1
14 12 11 9 7 6 2
```

se va afișa pe ecran

4

2. (5p) Se consideră un dicționar de cuvinte  $d$  și un șir  $s$  fără spații. Fiecare cuvânt din dicționar are asociată o pondere. Să se obțină în câte moduri poate fi  $s$  partiționat în cuvinte din dicționar prin adăugarea de spații astfel încât suma ponderilor cuvintelor care formează partiția  $s$  să fie **minimă**.

Se citește din fișierul "intrare.txt" un număr natural  $N$ , urmat de șirul  $S$ , urmat din  $N$  linii, reprezentând cuvintele din dicționar și ponderile acestora. Se va afișa pe ecran costul partiționării optime și numărul modurilor în care se poate obține această împărțire. Dacă  $s$  nu poate fi partiționat se va afișa -1.

Complexitate  $O(n^2)$

intrare.txt	iesire
8 tehnicideoptimizare tehnic 3 ide 2 nici 1 tehnici 5 de 5 optimi 1 zare 1 optimizare 2	7 2

tehnic + ide + optimizare --- cost 7

tehnic + ide + optimi + zare --- cost 7

tehnici + de + optimizare --- cost 12

### Observatii.

1. Se acordă **1p** din oficiu. Nota reprezintă 30% din nota finală
2. Se va preda un director/fișiere cu numele **grupa\_Nume\_Prenume\_Nr** (se pot pune în director proiectul sau doar fișiere sursă, se pot rezolva cerințele 1 și 2 în metode/clase/fișiere separate)
3. Se va scrie în comment o adresă de email pentru primirea rezultatelor