

## Subiecte Teză Semestrul I

**Se acordă 2 puncte din oficiu**

### Partea I. Backtracking - 3 puncte( 6\*0.5 puncte )

6 subiecte din următoarele

<http://panseluta.ml/c/Backtracking/Probleme%20Backtracking.html>

### Partea II. Greedy - 2 puncte

1 problemă rezolvată în C++ din următoarele

1. Se introduce de la tastatură o mulțime de numere pozitive,  $P$  și un număr  $M$ . Se cere să se afișeze la ecran o submulțime a lui  $P$  a cărui sumă a elementelor să fie cel mult  $M$ .
2. Se introduce de la tastatură o mulțime  $X = \{ x_1, x_2, \dots, x_n \}$  cu elemente reale. Se cere să se afișeze la ecran o submulțime a lui  $X$  astfel încât suma elementelor submulțimii să fie minimă.
3. Se introduce de la tastatură numărul natural  $k > 1$ . Se cere să se afișeze la ecran cel mai mic număr natural  $n$  având exact  $k$  divizori naturali proprii (diferiți de 1 și  $n$ ).

### Partea III Grafuri neorientate 3 puncte ( 2\*1,5 puncte )

2 probleme rezolvate în C++ din următoarele:

1. Se dă lista muchiilor unui graf neorientat. Să se afișeze matricea de adiacență a grafului.

#### **Date de intrare**

Fișierul de intrare *adiacenta.in* conține pe prima linie numerele  $n$  și  $m$ , reprezentând numărul de vârfuri ale grafului și numărul de muchii date în continuare. Fiecare dintre următoarele  $m$  linii conține câte o pereche de numere  $i, j$ , cu semnificația că există muchie între  $i$  și  $j$ .

#### **Date de ieșire**

Fișierul de ieșire *adiacenta.out* va conține  $n$  linii, pe fiecare linie fiind câte  $n$  valori separate prin exact un spațiu, reprezentând matricea de adiacență a grafului dat.

#### **Restricții și precizări**

$$1 \leq n \leq 100$$

$$1 \leq i, j \leq n$$

muchiiile se pot repeta în fișierul de intrare

#### **Exemplu**

*adiacenta.in*

5 8

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

**adiacenta.out**

0 1 1 1 0  
 1 0 0 1 0  
 1 0 0 1 1  
 1 1 1 0 1  
 0 0 1 1 0

2. Se dă lista muchiilor unui graf neorientat. Să se afișeze, pentru fiecare vârf al grafului, lista vecinilor săi.

**Date de intrare**

Fișierul de intrare *listavecini.in* conține pe prima linie numărul  $n$ , reprezentând numărul de vârfuri ale grafului. Fiecare dintre următoarele linii conține câte o pereche de numere  $i, j$ , cu semnificația că există muchie între  $i$  și  $j$ .

**Date de ieșire**

Fișierul de ieșire *listavecini.out* va conține  $n$  linii. Fiecare linie  $i$  va conține numărul de vecini ai vârfului  $i$  urmat de aceștia, în ordine crescătoare.

**Restricții și precizări**

$$1 \leq n \leq 100$$

$$1 \leq i, j \leq n$$

muchiiile se pot repeta în fișierul de intrare dar se va considera una singură

**Exemplu****listavecini.in**

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

**listavecini.out**

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

3. Dându-se o matrice de numere întregi cu  $n$  linii și  $n$  coloane, să se verifice dacă este sau nu matrice de adiacență asociată unui graf neorientat.

**Date de intrare**

Programul citește de la tastatură numărul  $n$ , iar apoi se citesc de pe  $n$  linii câte  $n$  numere naturale, separate prin spații, reprezentând elementele matricei.

**Date de ieșire**

Programul va afișa pe ecran **DA** dacă matricea este de adiacență, sau **NU** dacă nu este matrice de adiacență.

**Restricții și precizări**

$$3 \leq n \leq 100$$

Un graf neorientat nu admite nici bucle (deci nu există muchie de la un nod la el însuși) și nici muchii multiple între aceleași două noduri.

**Exemplul 1:****Intrare**

```
4
0 1 1 0
1 0 0 0
1 0 0 1
0 0 1 0
```

**Ieșire**

DA

**Exemplul 2:****Intrare**

```
4
0 1 1 0
1 0 0 0
0 0 0 1
0 0 0 0
```

**Ieșire**

NU

4. Se dă lista muchiilor unui graf neorientat. Să se afișeze vârfurile izolate ale grafului.

**Date de intrare**

Fișierul de intrare **grade.in** conține pe prima linie numărul  $n$ , reprezentând numărul de vârfuri ale grafului. Fiecare dintre următoarele linii conține câte o pereche de numere  $i, j$ , cu semnificația că există muchie între  $i$  și  $j$ .

**Date de ieșire**

Fișierul de ieșire **grade.out** va conține pe prima linie  $n$  numere naturale, reprezentând gradele vârfurilor, în ordinea vârfurilor.

**Restricții și precizări**

$$1 \leq n \leq 100$$

$$1 \leq i, j \leq n$$

muchiiile se pot repeta în fișierul de intrare dar se va considera una singură

**Exemplu****grade.in**

```
5
1 4
1 3
3 5
4 5
2 4
1 2
```

4 2

3 4

**grade.out**

3 2 3 4 2

5. Se dă lista muchiilor unui graf neorientat. Să se afișeze vârfurile de grad maxim.

**Date de intrare**

Fișierul de intrare **gradmax.in** conține pe prima linie numărul  $n$ , reprezentând numărul de vârfuri ale grafului. Fiecare dintre următoarele linii conține câte o pereche de numere  $i, j$ , cu semnificația că există muchie între  $i$  și  $j$ .

**Date de ieșire**

Fișierul de ieșire **gradmax.out** va conține pe prima linie numărul  $m$  de vârfuri de grad maxim, urmat de cele  $m$  vârfuri de grad maxim, în ordine crescătoare, separate prin exact un spațiu.

**Restricții și precizări** $1 \leq n \leq 100$  $1 \leq i, j \leq n$ 

muchii se pot repeta în fișierul de intrare dar se va considera una singură

**Exemplu****gradmax.in**

5

1 4

2 5

2 3

2 1

4 5

3 2

4 3

**gradmax.out**

2 4

# BAFTĂ!