

## **Proiectarea Algoritmilor - Tema 3**

**Data publicare: 2 mai 2013**

**Deadline: 19 mai 2013 - 23:55**

Mihai Bivol  
Sorina Sandu  
Emma Sevastian

## 1 Problema 1

Dupa cum știm, Ion și Vasile sunt foarte pasionați de fotbal. După ce Vasile l-a bătut de multe ori pe Ion la fotbal pe hârtie, Ion s-a decis să se dea învins și ca să îi arate lui Vasile că nu îi poartă pică, îl invită la următorul meci de fotbal. Cum Vasile e fan înfocat, e mereu la zi cu rezultatele meciurilor.

Din păcate, el nu este prea bun la calcule, așa că vă roagă pe voi să îi găsiți echipele care nu mai au cum să câștige cupa, astfel încât meciul pe care îl vor vedea să aibă cel puțin o echipă posibil câștigatoare a campionatului.

În liga pe care cei doi prieteni o urmăresc, regulile sunt puțin diferite față de fotbalul pe care îl cunoaștem. O echipă va primi 2 puncte pe un meci câștigat și 1 punct pe remiză. În plus, pot exista mai mult de două meciuri între două echipe date și ordinea în care meciurile sunt jucate este total aleatoare.

### 1.1 Cerință

Știind echipele, toate meciurile jucate până la un moment dat și toate meciurile ce au mai rămas de jucat, trebuie să răspundeți la un număr de întrebări de tipul "Poate echipa X să câștige campionatul?".

O echipă are șanse de a câștiga titlul dacă există o serie de rezultate ale meciurilor viitoare care o vor plasa pe primul loc al clasamentului. Presupunând că echipa va obține doar victorii, va trebui să verificați dacă există o astfel de configurație.

### 1.2 Format date intrare / ieșire

Executabilul generat va citi datele de intrare din fișierul fotbal.in și va afișa rezultatul în fișierul fotbal.out .

#### 1.2.1 Intrare

Pe **prima linie** a fișierului de intrare se vor afla **4** numere naturale:

- **N** - Numărul de echipe din campionat
- **P** - Numărul de partide jucate
- **K** - Numărul de etape ale campionatului (o echipă X va juca cu fiecare din celelalte  $N - 1$  echipe K meciuri)
- **Q** - Numărul de întrebări

Pe **următoarele P linii** se vor afla meciurile jucate în următorul format:

INDEX\_ECHIPA\_1 INDEX\_ECHIPA\_2 REZULTAT

unde:

- INDEX\_ECHIPA\_1 și INDEX\_ECHIPA\_2 sunt numere naturale în intervalul  $[0, N)$ .

- **REZULTAT** poate fi:

- **WIN**: echipa cu indexul `INDEX_ECHIPA_1` a câștigat meciul și a primit cele 2 puncte.
- **DRAW**: remiză, fiecare echipă a primit câte 1 punct.

Pe fiecare din următoarele **Q linii** se va afla câte un număr  $X_i, 0 \leq X_i < N$ , reprezentând o întrebare de tipul: “Poate echipa  $X_i$  câștiga campionatul?”.

### 1.2.2 Ieșire

În fișierul de ieșire se vor afla **Q linii**. Dacă echipa  $X_i$  poate câștiga campionatul, pe linia  $i$  se va afla șirul **TRUE**. Altfel, pe linia  $i$  se va afla șirul **FALSE**.

## 1.3 Limite

$$\begin{aligned} 2 &< N \leq 50 \\ 0 &< P \leq K * \frac{N*(N-1)}{2} \\ 0 &< K \leq 30 \\ 1 &< Q \leq 20 \end{aligned}$$

### 1.3.1 Exemple

#### Exemplul 1

```
fotbal.in
3 2 2 1
0 1 WIN
1 2 DRAW
1
```

```
fotbal.out
TRUE
```

Sunt 3 echipe: 0, 1, 2

După cele două meciuri jucate clasamentul este:

0 - 2p

1 - 1p

2 - 1p

Mai sunt de jucat meciurile:

0 - 1, 0 - 2, 0 - 2, 1 - 2

Vrem să știm dacă echipa 1 poate câștiga campionatul. Răspunsul este **TRUE**. Echipa 1 câștigă toate meciurile și va avea în total 5p. Echipa 0 Va face remiză cu 2 și va avea în total 4p. Echipa 2 va avea în total 3p.

#### Exemplul 2

```
fotbal.in
3 3 2 1
```

0 1 WIN  
 2 1 WIN  
 0 1 DRAW  
 1

fotbal.out  
 FALSE

Sunt 3 echipe: 0, 1, 2  
 După cele două meciuri jucate clasamentul este:  
 0 - 3p  
 2 - 2p  
 1 - 1p

Mai sunt de jucat meciurile:  
 0 - 2, 0 - 2, 1 - 2

Vrem să știm dacă echipa 1 poate câștiga campionatul. Răspunsul este **FALSE**  
 În cel mai bun caz pentru 1. Echipa 1 câștigă meciul rămas cu 2 și va avea în total 3 puncte. Dacă echipa 0 face cel puțin o remiză, se va clasa înaintea lui 1. Dacă echipa 0 pierde toate meciurile, atunci echipa 2 va fi echipa care va câștiga campionatul.

#### 1.4 Hint-uri flux

O rezolvare posibilă este calcularea unui flux maxim. Pentru aceasta, va trebui să ne reprezentăm problema sub forma unui graf și să asociem muchiilor capacități corespunzătoare.

Când este o echipă eliminată matematic? Când punctajul maxim pe care îl poate obține va fi întrecut de cel puțin una din echipele adverse.

Fie  $S$  mulțimea echipelor. Oricare ar fi  $X \in S$  o echipă, fie:

- $puncte_X$  - numărul de puncte ale echipei  $X$
- $maxp_X$  - maximul de puncte pe care echipa  $X$  îl poate obține până la sfârșit.
- $g_{x,y}$  - numărul de jocuri pe care  $x$  trebuie să le mai joace împotriva lui  $y$

$X$  este eliminată matematic dacă există  $T \subset S$  astfel încât:

$$\sum_{x \in T} (puncte_x) + 2 * \sum_{x,y \in T} (g_{x,y}) > maxp_X * |T|$$

Altfel spus, dacă există un subset  $T$  de echipe pentru care media aritmetică a punctajelor adunate de echipele din  $T$  este mai mare decât numărul maxim de puncte pe care le poate obține  $X$ , atunci  $X$  este eliminată.

## 2 Problema 2

Dupa ce au găsit bilete la meciul dorit, Ion și Vasile s-au gândit să se joace ceva. Întrucât nu au crezut că o să aleagă meciul așa de repede, nu au luat niciun joc cu ei. Așa că acum, ei au la dispoziție doar o tablă de dimensiune  $n * m$  și un zar. Tot încercând să găsească un joc, s-au gândit la următoarea problemă:

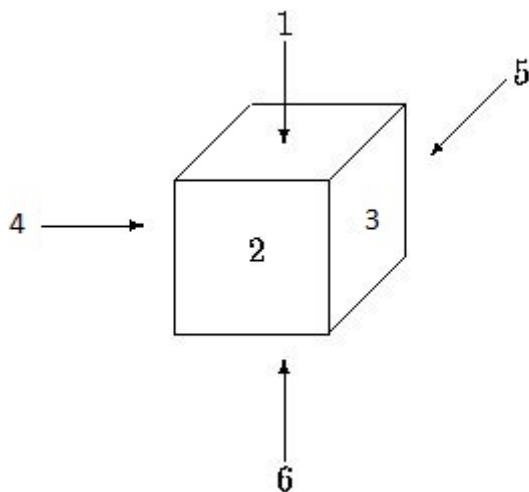
### 2.1 Cerința

Având o tablă de dimensiune  $n * m$  și un zar (care pe fiecare față are un număr între 1 și 100) se cere aflarea sumei minime (după numărul afișat pe fața superioară a cubului) între o sursă și o destinație prin rotirea zarului pe tablă. Zarul poate fi rotit pe oricare parte.

#### Observații:

- zarul va porni din poziția sursă cu fața 1 în sus, ca în figură.
- poziția  $(0, 0)$  se află în colțul din stânga-sus iar  $(n-1, m-1)$  în dreapta-jos.

Numerotare fețe zaruri :



### 2.2 Format date intrare / ieșire

Executabilul generat va citi datele de intrare din fișierul zar.in și va afișa rezultatul în fișierul zar.out

#### 2.2.1 Intrare

Pe prima linie a fișierului de intrare se vor afla 2 numere naturale:

- N - Numărul de linii ale tablei
- M - Numărul de coloane ale tablei

A **doua linie** va conține cele 6 valori ale fețelor cubului:

```
VALOARE_FAȚĂ_1 VALOARE_FAȚĂ_2 VALOARE_FAȚĂ_3 VALOARE_FAȚĂ_4 VALOARE_FAȚĂ_5  
VALOARE_FAȚĂ_6
```

Următoarele **2 linii** vor conține sursa respectiv destinația:

```
SURSA_LINIE SURSA_COLOANĂ  
DESTINAȚIE_LINIE DESTINAȚIE_COLOANĂ
```

### 2.2.2 Ieșire

În fișierul de ieșire se va afișa suma minimă obținută.

## 2.3 Limite

$$2 < N \leq 300$$

$$2 < M \leq 300$$

Valorile fețelor cubului sunt pozitive și nu vor depăși 100

### 2.3.1 Exemplu

```
zar.in  
4 4  
1 2 3 4 5 6  
0 0  
3 3
```

```
zar.out  
22
```

#### Explicație

O variantă posibilă de drum este:

$(0, 0) - (1, 0) - (2, 0) - (2, 1) - (2, 2) - (2, 3) - (3, 3)$

$$1 + 5 + 6 + 4 + 1 + 3 + 2 = 22$$

## 3 Punctare

Punctajul pentru cele două probleme este de 40 de puncte fiecare. 10 puncte vor fi acordate pentru coding-style, 10 puncte pentru comentarii și README. Pentru detalii puteți să vă uitați și peste regulile generale de trimitere a temelor.

Vor exista în total 20 de teste de 4 puncte fiecare, 10 teste pentru prima problemă, și 10 teste pentru cea de-a doua.

Punctele pe teste sunt independente, punctajul pe un anumit test nefiind condiționat de alte teste. Fiecare problemă va avea o limită de timp pe test (precizată mai jos) - dacă execuția programului pe un test al acelei probleme va dura mai mult decât limita de timp veți primi automat 0 puncte pe testul respectiv și execuția va fi întreruptă.

În fișierul README va trebui să descrieți soluția pe care ați ales-o pentru fiecare problemă, să precizați complexitatea pentru fiecare și alte lucruri pe care le considerați util de menționat.

Corectorii își rezervă dreptul de a scădea puncte dacă vor considera acest lucru necesar.

## 4 Format arhiva și testare

Temele pot fi testate automat pe vmchecker - acesta suportă temele rezolvate în C/C++ și Java. Dacă doriți să realizați tema în alt limbaj trebuie să trimiteți un e-mail lui Traian Rebedea (traian.rebedea@cs.pub.ro) în care să îi cereți explicit acest lucru.

Arhiva cu rezolvarea temei trebuie să fie .zip și să conțină:

- Fișierul/fișierele sursă
- Fișierul Makefile
- Fișierul README

Fișierul pentru make trebuie denumit **obligatoriu** Makefile și trebuie să conțină următoarele reguli:

- build, care va compila sursele și va obține executabilele.
- run-p1, care va rula executabilul pentru problema 1
- run-p2, care va rula executabilul pentru problema 2
- clean, care va șterge executabilele generate.

**Atentie!** Numele regulilor trebuie să fie exact cele de mai sus.

Limitele de timp pentru problema 1 sunt:

- C/C++ 2 secunde
- Java 2 secunde

Limitele de timp pentru problema 2 sunt:

- C/C++ 2 secunde
- Java 2 secunde