

# Tema 3 – Grafuri

Responsabili: Ana-Maria SIMION, Maria-Anca BĂLUȚOIU,  
Emanuel TERTEȘ, Dumitru-Pavel BEȚIU

Data postării: 14.05.2024

**Deadline: 26.05.2024 ora 23:59**

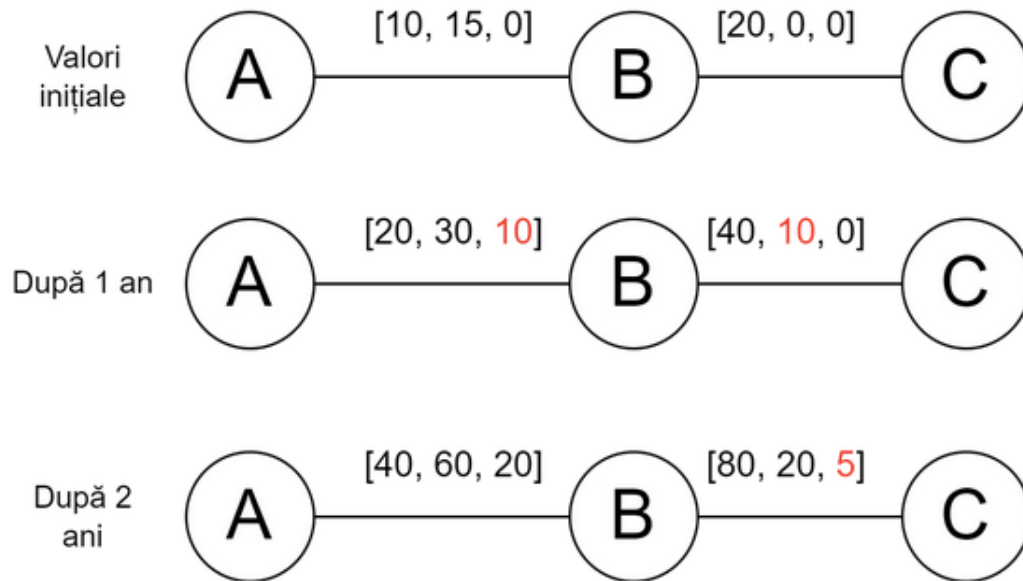
## 1 Descriere

Ne aflăm la începutul secolului al XIX-lea în Europa. Marea Britanie a construit deja primele căi ferate moderne și succesul rutei Liverpool-Manchester a determinat și celelalte națiuni să construiască un sistem modern de căi ferate. Astfel, toate țările au căzut de acord să organizeze un congres unde să ia toate deciziile importante legate de noul sistem de transport care urmează să împânzească tot continentul. După multe întâlniri și dezbateri, națiunile au început să întocmească planul pentru noul sistem feroviar, însă au întâmpinat niște probleme pentru care au nevoie de ajutorul tău.

## 2 Cerința 1

Congresul a hotărât să încerce să păstreze câteva părți din căile ferate vechi, însă trebuie să analizeze foarte bine ce rute ar mai putea fi păstrate și ce ar trebui înlocuit. Fiecare drum care unește 2 orașe este reprezentat în planul căii ferate sub forma de ni tronsoane, unde  $i$  este indexul rutei în lista de  $R$  rute. Congresul vrea să facă o analiză exactă a gradului de uzură pe care îl va avea fiecare bucată de tronson peste  $K$  ani. Fiecare bucată de tronson din fiecare rută a fost analizată și i s-a alocat un grad  $D$  de uzură. Dacă tronsonul este într-o stare perfectă gradul de uzură este 0% și se consideră stricat când ajunge la 100% (100% este gradul maxim la care poate ajunge o bucată de tronson, dar aceasta nu dispăre). Știind că fiecare tronson uzat afectează bucățile vecine și că gradul de uzură se dublează la trecerea fiecărui an, trebuie să modelăm cum va arăta planul vechilor căi ferate peste  $K$  ani. Dacă o bucată de tronson afectată se află lângă o bucată de tronson într-o stare perfectă, după un an aceasta o să ia gradul de uzură/2 al tronsonului vecin, dacă există 2 tronsoane vecine afectate se folosește valoarea mai mare. De asemenea, trebuie să extragem o listă cu indicii rutelor care merită păstrate. Se consideră că o rută merită păstrată dacă media gradului de uzură pe toată ruta este mai mică ca  $L$ , acest lucru se decide doar după ce au trecut  $K$  ani. Pentru graful de uzură se va folosi tipul de date **float** și precizie de 2 zecimale. Vor exista input-uri în care graful nu este conex. Graful este neorientat.

Exemplu de actualizare în timp a gradelor de degradare ale tronsoanelor inițial neafectate (cu roșu sunt evidențiate actualizările valorilor de degradare):



**Explicație:** Ultimul tronson de pe ruta A-B are ca vecini un tronson cu degradare 15% și un tronson cu degradare 20%, aflat pe ruta B-C. Având în vedere că valoarea cea mai mare dintre cele menționate este 20%, vom actualiza tronsonul curent la valoarea de 10% ( $20 / 2$ ). Cel de-al doilea tronson al rutei B-C are ca vecini un tronson cu degradare 20% și unul cu degradare 0%, prin urmare tronsonul curent va avea o degradare de 10% în anul următor.

## 2.1 Formatul fișierelor

Fișierul de intrare va avea următorul format:

R (numărul de rute)  
 K (numărul de ani)  
 L (gradul de uzură acceptabil)  
 [Oraș1, Oraș2, Număr de tronsoane, [Lista cu gradele de afectare pe fiecare tronson]]

### Exemplu

#### Input:

```
4
3
40
Lisabona Madrid 3 10 15 0
Madrid Pamplona 3 0 0 0
Pamplona Brest 4 0 20 40 0
Venetia Zagreb 2 0 0
```

#### Output:

```
Lisabona Madrid 3 80.00 100.00 30.00
Madrid Pamplona 3 7.50 2.50 10.00
Pamplona Brest 4 40.00 100.00 100.00 80.00
Venetia Zagreb 2 0.00 0.00
2 4
```

#### Explicatie:

##### • An1:

```
Lisabona, Madrid, 3, [20%, 30%, 7.5%]
Madrid, Pamplona, 3, [0%, 0%, 0%]
Pamplona, Brest, 4, [10%, 40%, 80%, 20%]
Venetia, Zagreb, 2, [0%, 0%]
```

- An2:

Lisabona, Madrid, 3, [40%, 60%, 15%]  
Madrid, Pamplona, 3, [3.75%, 0%, 5%]  
Pamplona, Brest, 4, [20%, 80%, 100%, 40%]  
Venetia, Zagreb, 2, [0%, 0%]

- An3:

Lisabona, Madrid, 3, [80%, 100%, 30%]  
Madrid, Pamplona, 3, [7.5%, 2.5%, 10%]  
Pamplona, Brest, 4, [40%, 100%, 100%, 80%]  
Venetia, Zagreb, 2, [0%, 0%]

### 3 Cerința 2

Pentru această cerință, pornim de la o hartă ce conține  $n$  orașe între care există  $m$  căi ferate. Considerăm că o cale ferată este bidirecțională și definită pe baza a două orașe, cele între care apare calea ferată, și are o anumită distanță. Deoarece costă prea mult întreținerea pentru cele  $m$  căi ferate, noi dorim să renunțăm la unele astfel încât să rămânem cu cel mult  $k$  căi ferate. Renunțarea la unele căi ferate ar putea impacta profitabilitatea rețelei de transport. Vom considera că profitabilitatea rețelei de transport este influențată de numărul de orașe pentru care costul drumului minim, care pornește din orașul de start furnizat, rămâne nemodificat după eliminarea unor căi ferate. Scopul nostru constă în păstrarea în rețeaua de transport a cel mult  $k$  căi ferate astfel încât profitabilitatea rețelei să fie maximă (conține numărul maxim de orașe pentru care nu se modifică drumul de cost minim de la orașul de start).

Rezultatul constă în determinarea numărului de căi ferate pe care le putem opri și care sunt aceste căi ferate. Căile ferate vor fi afișate în ordinea în care ele au apărut în fișierul de intrare.

Dacă vor exista mai multe variante prin care putem selecta același număr de căi ferate, le vom alege pe cele care ne asigură drumuri minime nemodificate către cele mai apropiate orașe.

#### 3.1 Formatul fișierelor

Fișierul de intrare va avea următorul format:

Oraș start  
K (numărul maxim de căi ferate care pot fi păstrate)  
M (numărul căilor ferate existente)  
[M căi ferate cu formatul: Oraș1 Oraș2 distanță]

Fișierul de ieșire va avea următorul format:

T (numărul căilor ferate care vor fi păstrate)  
[T căi ferate cu formatul: Oraș1 Oraș2]

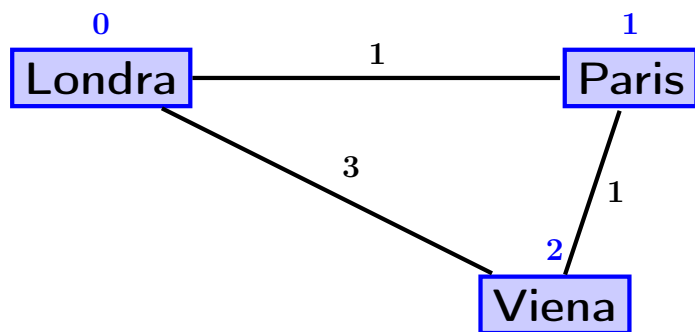
### Exemplul 1

Input:

Londra  
2  
3  
Londra Paris 1  
Paris Viena 1  
Viena Londra 3

Output:

2  
Londra Paris  
Paris Viena



#### Explicație

Drumul minim de la Londra la Paris este de lungime 1, iar costul drumului minim de la Londra la Viena este 2. Dacă păstrăm primele 2 căi ferate, aceste costuri nu se modifică. Calea ferată de la Viena la Londra nu era folosită pentru niciun drum minim.

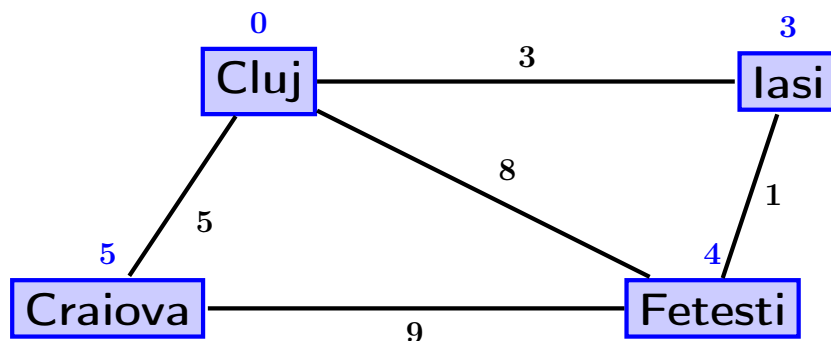
### Exemplul 2

Input:

Cluj  
2  
5  
Iasi Cluj 3  
Craiova Fetesti 9  
Craiova Cluj 5  
Cluj Fetesti 8  
Iasi Fetesti 1

Output:

2  
Iasi Cluj  
Iasi Fetesti



#### Explicație

Drumul minim de la Cluj la Iasi are costul 3, drumul minim de la Cluj la Fetesti are costul 4 și drumul minim de la Cluj la Craiova are costul 5. Dacă vom păstra căile ferate (Iasi, Cluj) și (Iasi, Fetesti) vom menține drumurile minime pentru Iasi și Fetesti.

## 4 Testarea temei

Temele trebuie să fie încărcate pe vmchecker.

**NU** se acceptă teme trimise pe e-mail sau altfel decât prin intermediul vmchecker-ului.

O rezolvare constă într-o arhivă de tip **zip** care conține toate fișierele sursă alături de un Makefile, ce va fi folosit pentru compilare, și un fișier **README**, în care se vor preciza detaliile implementării.

**Makefile**-ul trebuie să aibă obligatoriu regulile pentru **build** și **clean**. Regula **build** trebuie să aibă ca efect compilarea surselor și crearea binarului **tema3**. Datele se citesc din fișierul **tema3.in** și rezultatele se vor scrie în fișierul **tema3.out**.

Programul vostru va primi, ca argumente în linia de comandă, 1 sau 2 pentru a indica cerința rezolvată.

## 5 Punctaj

O temă perfectă valorează 100 de puncte. 95 de puncte se vor acorda pentru teste și 5 puncte pentru README.

În urma corectării manuale, punctajul obținut pe vmchecker poate fi scăzut cu maxim 15 puncte pentru coding style. De asemenea, conținutul fișierului README va fi verificat manual, iar punctajul obținut pentru README poate fi diminuat.

Punctajul pe teste este următorul:

Cerința	Punctaj
Cerința 1	45 puncte
Cerința 2	50 puncte
README	5 puncte
BONUS (testat cu valgrind)	20 puncte

### Atenție!

- Orice rezolvare care nu conține structurile de date specificate **NU** este punctată.
  - Este obligatoriu ca graful să fie reprezentat folosind liste de adiacență!
- Temele vor fi punctate doar pentru testele care sunt trecute pe vmchecker.
- Nu lăsați warning-urile nerezolvate, deoarece veți fi depunțați.
- Tema este individuală! Toate soluțiile trimise vor fi verificate, folosind o unealtă pentru detectarea plagiatului.