

Tema 3 – Grafuri

Responsabili: Maria-Anca BĂLUȚOIU, Andrei DUGĂEȘESCU,
Ioana-Theodora POPA, Ana-Maria SIMION

Data postării: 05.05.2023

Deadline: 21.05.2023 ora 23:59

1 Descriere

Ne aflăm în anul 1783 în Bristol și Jim Hawkins intră în posesia unei hărți vechi a orașului și a unui mesaj ce conține indicații menite să-l conducă la o insulă plină de comori. Pe hartă sunt marcate cele mai importante obiective de la acea vreme din oraș. Mesajul care însoțește harta ne spune că "*Orașul Bristol este unul dintre cele mai vechi orașe din Regatul Unit, cu o istorie bogată și îndelungată. Orașul a început să se dezvolte într-un mod organizat în jurul secolului al XI-lea. În acea perioadă, Bristol-ul era un important centru comercial datorită poziției sale strategice de-a lungul râului Avon. Astfel era și un loc frecventat de pirați. Transportul maritim era principala modalitate de a aduce mărfuri în oraș și locuitorii, în special proprietarii de magazine și ateliere, au fost constrânși să găsească o rută cât mai convenabilă pe care să transporte marfa. Ruta trebuia să conțină toate punctele specificate pe harta și să fie minimă. Pentru a găsi următorul indiciu care ne va conduce la comoară, trebuie să găsim ruta de transport folosită în acea perioadă și costul ei. În cazul în care nu putem ajunge la toate obiectivele pe uscat, va trebui să stabilim câte o rută principală și costul ei minim pentru fiecare grup în parte.*"

2 Cerință 1

Pentru această cerință, harta este reprezentată sub forma unui graf neorientat. Obiectivele din oraș reprezintă nodurile grafului, iar muchiile sunt echivalentul unor drumuri bidirecționale care există între aceste noduri. Pentru fiecare drum cunoaștem o distanță. Un exemplu de astfel de graf este prezentat în Figura 1.

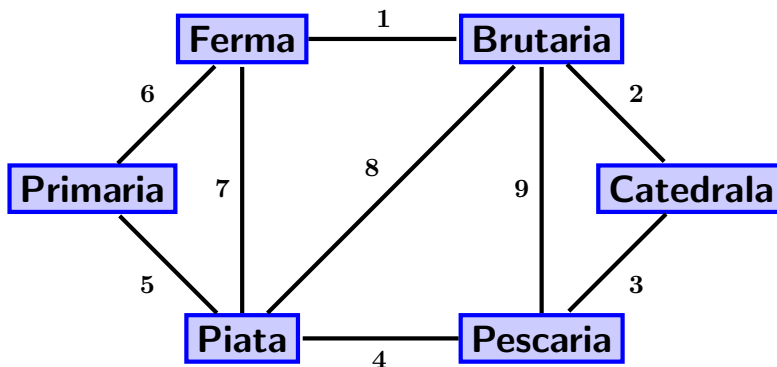


Figure 1: Exemplu de hartă toate locurile din oraș sunt situate pe aceeași bucată de uscat

În cazul acestui prim exemplu, toate obiectivele din oraș sunt situate pe aceeași bucată de uscat. Cu alte cuvinte, indiferent de obiectivul de la care pornim vom putea ajunge în oricare alt obiectiv din oraș.

Există însă situații în care harta este împărțită în mai multe zone care sunt despărțite de ape. Un astfel de exemplu este prezentat în Figura 2. După cum putem observa, de această dată avem două zone (fiecare zonă fiind colorată diferit).

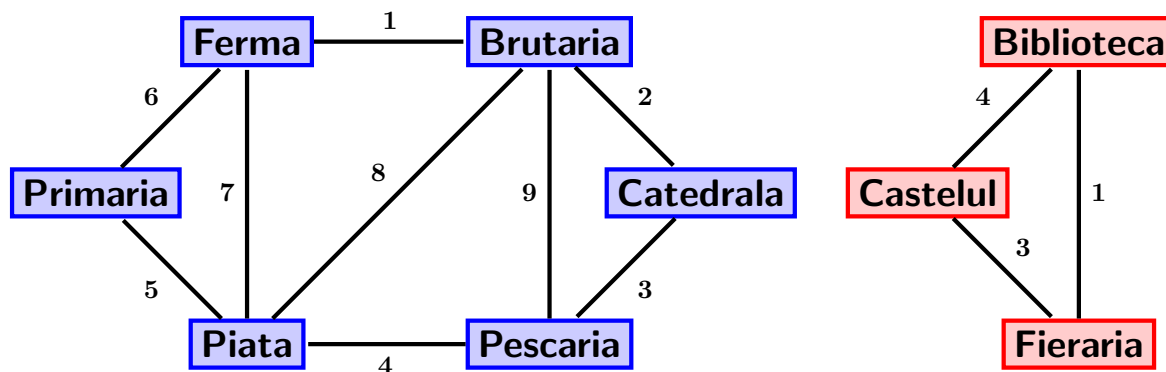


Figure 2: Exemplu de hartă în care obiectivele din oraș sunt împărțite de râuri

Voi va trebui să-l ajutați pe Jim Hawkins să determine pentru o hartă dată numărul de zone care sunt despărțite de ape.

După ce determinați numărul de zone, va trebui ca pentru fiecare zonă să îl ajutați pe Jim Hawkins să determine ce drumuri ar trebui să renoveze astfel încât costul total al acestora să fie minim și indiferent de ce obiectiv alegem să putem ajunge în oricare altul aflat pe aceeași bucată de uscat cu el parcurgând doar drumuri renovate.

În Figura 3 sunt evidențiate drumurile care ar trebui renovate pentru harta de la Figura 2.

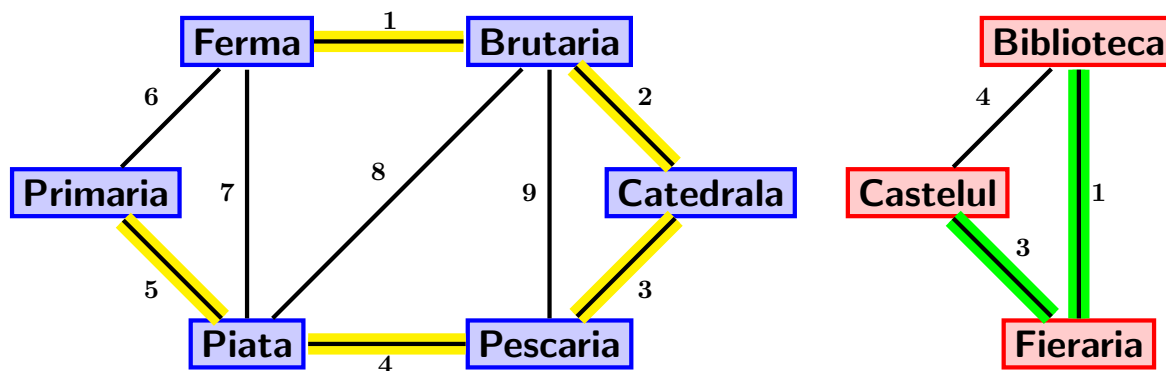


Figure 3: Evidențierea drumurilor ce ar trebui să fie renovate

Voi va trebui să afișați costul total al drumurilor ce urmează să fie renovate pentru fiecare zonă despărțită de ape în **ordine crescătoare**.

Astfel, pentru cerința 1 veți avea de rezolvat două sarcini:

- a) Determinarea numărului de zone care sunt despărțite de ape.
- b) Pentru fiecare zonă despărțită de ape, trebuie să determinați costul total minim al drumurilor ce ar trebui să fie renovate.

2.1 Formatul fișierelor

Fișierul de intrare

- Pe prima linie din fișierul de intrare vor exista două numere N și M , iar N reprezintă numărul de obiective aflate pe hartă și M reprezintă numărul de drumuri bidirecționale care există între aceste obiective.
- Pe următoarele M linii vom avea descrise drumurile bidirecționale. Fiecare astfel de linie va conține două șiruri de caractere și un număr natural (cost), despărțite printr-un spațiu. Spre exemplu:

Ferma Brutaria 1

- Numele fișierului din care se citesc datele este "tema3.in".

Observație

Numele obiectivelor din oraș va fi format dintr-un singur cuvânt.

Fișierul de ieșire

- Pe prima linie se va scrie un număr natural K reprezentând numărul de zone care sunt despărțite de ape.
- Pe următoarele K linii, vom scrie câte un număr natural $cost_total[i]$ ce reprezintă costul total al drumurilor ce trebuie să fie renovate pentru zona i .

Observație

Aceste valori sunt afișate în **ordine crescătoare!**

2.2 Exemplu

tema3.in

```
9 12
Ferma Brutaria 1
Brutaria Catedrala 2
Catedrala Pescaria 3
Pescaria Piata 4
Primaria Piata 5
Primaria Ferma 6
Ferma Piata 7
Piata Brutaria 8
Brutaria Pescaria 9
Biblioteca Fieraria 1
Castelul Biblioteca 4
Fieraria Castelul 3
```

tema3.out

```
2
4
15
```

3 Cerință 2

Harta găsită ne conduce la insulă. Ajunși aici, trebuie să ancorăm nava la o distanță destul de mare de mal, pentru că adâncimea apei este prea mică în unele locuri. Astfel, echipajul nostru este nevoit să folosească o barcă pentru a căra comoara la corabie. Harta ne indică zonele prin care putem trece fără să declanșăm capcanele din jurul insulei. Sensul de trecere între două zone de pe hartă este specificat. Comoara este de mult timp ascunsă pe insulă, și între timp s-au format dune de nisip pe traseele indicate pe hartă. Înainte să înceapă încărcarea comorii pe corabie, echipajul analizează cu atenție drumul de la insula comorii la navă și marchează dunele de nisip. Fiecare nod are o valoare care reprezintă adâncimea apei în punctul respectiv. Cu cât încărcăm mai mult barca, cu atât se scufundă mai mult și nu va mai putea trece prin anumite zone. Pentru că timpul este prețios, iar echipajul este deja extenuat, trebuie să găsim cea mai bună variantă de a transporta comoara. Un membru al echipajului a venit cu o idee. Pe hartă, arcele reprezintă distanța dintre două puncte prin care putem trece, iar nodurile au atașate o valoare pentru adâncime. Membrul echipajului a determinat o formulă care să atribuie un scor unui arc pe baza distanței și a adâncimii. Folosim acest scor pentru a evalua fiecare alegere și, în final, să determinăm cea mai bună rută de urmat pentru a duce comoara de pe insulă la corabie. Odată aleasă ruta, aceasta nu poate fi schimbată. În mod ideal, căutăm rute cât mai scurte, pe care să putem căra cât mai mult din comoară. Formula folosită pentru evaluarea unui arc este:

$$\text{scor}(\text{arc}) = \frac{\text{distanță}(\text{arc})}{\text{adâncime}(\text{nodul_catre_care_se_duce_arcul})}$$

Observație

Tipul variabilei folosit pentru calculul scorului este **float**.

Important

Pentru rezolvarea acestei cerințe, veți porni de la **Algoritmul lui Dijkstra**.

3.1 Formatul fișierelor

Fișierul de intrare

- Pe prima linie din fișierul de intrare găsim două numere N și M . N reprezintă numărul de puncte aflate pe hartă care ne indică zonele prin care putem trece fără să declanșăm capcanele din jurul insulei. M reprezintă numărul de drumuri unidirecționale care există între aceste puncte.
- Pe următoarele M linii vom avea descrise drumurile unidirecționale. Fiecare astfel de linie va conține două șiruri de caractere și un număr natural (distanța), despărțite printr-un spațiu. Spre exemplu:

Insula Nod_2 5

- Pe următoarele N linii se citesc perechi formate din numele punctului de pe hartă urmat de adâncimea apei în punctul respectiv (număr natural).
- Pe ultima linie se citește o valoare care reprezintă greutatea comorii în kg (număr natural).

- În cazul nodurilor Insulă și Corabie adâncimea nu este folosită, aceasta o sa fie considerată o valoare implicită 1.

Fișierul de ieșire

- Toate datele afișate sunt numere naturale.
- Pe prima linie se vor scrie toate nodurile care constituie cel mai bun drum pentru a căra comoara de la insulă la corabie.
- Pe a doua linie se va afișa costul total al drumului.
- Pe a treia linie se va afișa greutatea maximă pe care o poate avea barca pentru a trece pe traseul găsit.
- Pe ultima linie se va afișa numărul de drumuri făcute pentru a căra întreaga comoară.
- De asemenea, trebuie să ne asigurăm că există cel puțin o rută prin care putem ajunge de la Corabie la Insulă, altfel vom afișa mesajul “Echipajul nu poate ajunge la insula” și problema s-a încheiat (nu putem încerca să găsim o rută de la Insulă la Corabie dacă nu putem ajunge la insulă în primul rând).
- Dacă găsim o rută de la Corabie la Insulă, însă nu există nicio rută de la Insulă la Corabie, se va afișa mesajul “Echipajul nu poate transporta comoara înapoi la corabie”.

3.2 Exemple

3.2.1 Exemplul 1

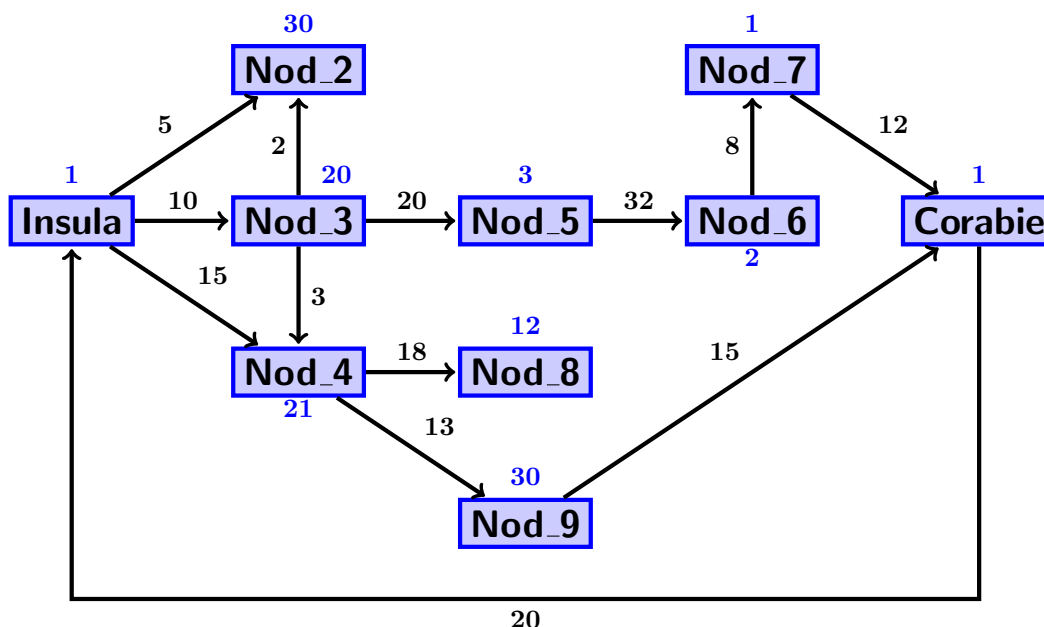


Figure 4: Harta de la exemplul 1 (pentru fiecare nod este precizată cu albastru adâncimea apei)

tema3.in

```
10 13
Insula Nod_2 5
Insula Nod_3 10
Insula Nod_4 15
Nod_3 Nod_2 2
Nod_3 Nod_5 20
Nod_3 Nod_4 3
Nod_4 Nod_8 18
Nod_4 Nod_9 13
Nod_5 Nod_6 32
Nod_6 Nod_7 8
Nod_7 Corabie 12
Nod_9 Corabie 15
Corabie Insula 20
Insula 1
Corabie 1
Nod_2 30
Nod_3 20
Nod_4 21
Nod_5 3
Nod_6 2
Nod_7 1
Nod_8 12
Nod_9 30
1200
```

tema3.out

```
Insula Nod_3 Nod_4 Nod_9 Corabie
41
20
60
```

Explicația

Există drum și de la corabie la insulă și de la insulă la corabie, ceea ce înseamnă că echipajul poate ajunge la insulă și poate căra comoara înapoi la bordul corabiei. Calculând raportul dintre distanță și adâncime, observăm că cel mai bun drum este Insula-Nod_3-Nod_4-Nod_9-Corabie. Costul acestui drum (suma arcelor) este 41. Cea mai mică adâncime de pe tot drumul este 20. Deci, la fiecare drum făcut între insulă și corabie putem căra maxim 20 kg. Greutatea totală a comorii este 1200 kg, deci trebuie să facem 60 de drumuri pentru a muta de pe insulă pe corabie întreaga comoară.

3.2.2 Exemplul 2

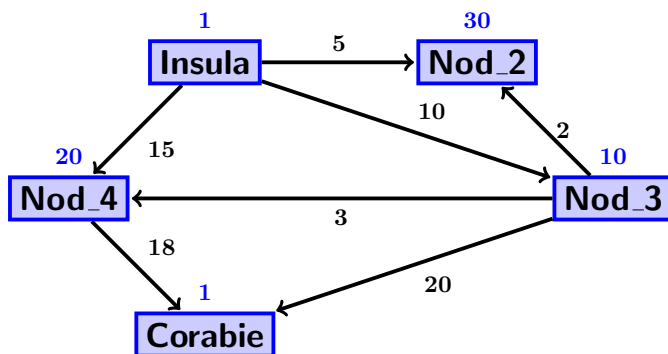


Figure 5: Harta de la exemplul 2 (pentru fiecare nod este precizată cu albastru adâncimea apei)

tema3.in

tema3.out

```
5 7
Insula Nod_2 5
Insula Nod_3 10
Insula Nod_4 15
Nod_3 Nod_2 2
Nod_3 Corabie 20
Nod_3 Nod_4 3
Nod_4 Corabie 18
Insula 1
Corabie 1
Nod_2 30
Nod_3 10
Nod_4 20
2540
```

Echipajul nu poate ajunge la insula

Explicație

Nu există drum de la Corabie la Insulă, deci echipajului îi este imposibil să ajungă la Insulă și să captureze comoara.

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. 80 de puncte se vor acorda pentru teste, 15 pentru coding style și 5 puncte pentru README. Vor exista atât teste simple, care verifică o operație specifică, dar și teste complexe, în care există majoritatea operațiilor.

Punctajul pe teste este următorul:

Cerința	Punctaj
Cerința 1 a)	15 puncte
Cerința 1 b)	25 puncte
Cerința 2	40 puncte
Coding style & Warning-uri	15 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.**