

Tema 2 - Hit the Road

Responsabili:

- Rares Cheseli
- Serban Ciofu

Deadline: 04 mai 2018, ora 23:55

Modificări și actualizări

* 13 aprilie

- modificare timeout valgrind

* 18 aprilie

- modificare afisare checkstyle
- precizare task 4
- prelungire deadline

* 25 aprilie

- modificare arhiva teste

Obiective

- aprofundarea cunostintelor in utilizarea limbajului C++
- Implementarea si utilizarea structurii de date hashtable(generica) pentru maparea numerelor de orase cu un index, si invers
- Modelarea unei harti rutiere sub forma unui graf orientat pentru o rezolvare eficienta a problemei
- Elaborarea unei rezolvari complexe la o cerinta inspirata din realitate

Intro

O harta rutiera, definita prin orase si autostrazi intre aceste orase, poate fi reprezentata sub forma unui graf orientat in care nodurile reprezinta orasele, iar muchiile reprezinta autostrazile (vom considera autostrazile ca fiind unidirectionale). O astfel de reprezentare a unei harti rutiere este foarte eficienta pentru a determina diferite proprietati ale acesteia (conectivitatea intre diferite orase, care sunt cele mai frecventate rute, cel mai scurt drum intre doua orase etc.).

Enunt

Fiind date orasele in ordinea lor lexicografica, legaturile dintre orase si orasele in care se afla benzinarii, va trebui sa determinati urmatoarele statistici si proprietati:

- cel mai aglomerat oras;
- validitatea harti rutiere;
- pentru un anumit oras, care sunt orasele inaccesibile lui;
- pentru o masina despre care se cunosc cantitatea initiala de combustibil, consumul acesteia si orasul in care se afla, trebuie sa determinati cea mai apropiata benzinarie in care poate alimenta (daca aceasta exista);
- fiind date traseele unor autobuze intre diferite orase, sa se determine traseul frecventat de cei mai multi calatori

Notiunile definite mai sus vor fi explicate in detaliu in sectiunea urmatoare.

Clarificari

Generale

- consideram lungimea oricarei autostrazi ca fiind 1 unitate;

 Search

- Catalog
- Reguli si Notare

Laboratoare

- Lab 1 - Introducere în C++
- Lab 2 - Bazele C++
- Lab 3 - Templates - ArrayList și LinkedList
- Lab 4 - Stivă și Coadă
- Lab 5 - Dicționar
- Lab 6 - Grafuri - Basics
- Lab 7 - Grafuri - Advanced
- Lab 8 - Arbori și Arbori Binari
- Lab 9 - ABC și Heap
- Lab 10 - Treap, AVL Tree, Red-Black Tree
- Lab 11 - Trie

Laboratoare din anii trecuti

- Laboratoare 2013
- Laboratoare 2014
- Laboratoare 2015
- Laboratoare 2016
- Laboratoare 2017

Teste practice

- Instructiuni test practic
- Simulare test practic
- Simulare test practic 2
- Depunctari simulare test practic
- Contact tutori
- Test practic 2h00 (cu bonus de timp)
- Test practic 1h30 (fara bonus de timp)
- Test practic restanta 30 Aug
- Test practic restanta 4 Sep

Teme

- Tema 1
- Tema 2
- Tema 3
- Regulament coding style

Resurse

- Instrumente Colaborative
- Aplicatie pentru telefonul mobil
- Debugging
- Data Structure Visualization

- consideram lungimea drumului minim intre doua orase ca fiind numarul minim de muchii care trebuie parcurse pentru a ajunge de la un oras la altul;

TASK 1

Se definește gradul de aglomerație al unui oras ca fiind numarul de autostrazi care intra in orasul respectiv. Voi va trebui sa determinati orasul care are gradul de aglomerație maxim. Daca exista mai multe orase cu grad maxim de aglomerație il vezi afisa pe cel mai mic din punct de vedere lexicografic.

TASK 2

Consideram ca o harta rutiera este valida daca exista un traseu(format din una sau mai multe muchii) intre oricare doua orase.

TASK 3

Veti primi o lista cu mai multe orase. Pentru fiecare oras din aceasta lista va trebui sa determinati o lista de orase cu care acesta nu poate comunica(doua orase nu pot comunica daca nu exista un drum dus-intors intre ele). Daca harta rutiera este valida, va exista cel putin un drum intre oricare doua orase. Lista oraselor inaccesibile pe care o determinati trebuie sortata lexicografic dupa nume. Daca nu exista orase inaccesibile(se poate ajunge in orice oras), afisati pe o singura linie mesajul "Exista drum intre oricare doua orase".

TASK 4

Pentru o lista de masini avand caracteristicile mentionate mai sus, trebuie sa determinati, pentru fiecare masina, cea mai apropiata benzinarie in care se poate ajunge cu masina respectiva. **Daca masina se afla intr-un oras cu benzinarie, aceasta va trebui sa ajunga la o benzinarie dintr-un oras diferit fata de cel din care porneste.** Pentru o masina aflata in orasul A si 2 benzinarii aflate in orasele B, respectiv C, consideram ca benzinaria B este mai apropiata de masina din A daca lungimea drumului minim de la A la B este mai mica decat lungimea drumului minim de la A la C.

Fie o masina cu X litri de combustibil initial si cu un consum de C litri per unitate. Pentru a putea parcurge o muchie dintre doua orase, masina TREBUIE sa aiba in rezervor cel putin (\geq) C litri de combustibil. X si C reprezinta numere intregi strict pozitive(strict mai mari ca 0).

TASK 5

Un traseu intre 2 orase este definit prin urmatorul tuplu: <nume_sursa nume_destinatie numar_calatori> unde:

- nume_sursa reprezinta numele orasului din care pleaca autobuzul
- nume_destinatie reprezinta numele orasului in care autobuzul opreste
- numar_calatori reprezinta cati calatori sunt in autobuz pe toata durata drumului

Autobuzul nu opreste in orase intermediare; opreste doar in orasul destinatie. Pot exista mai multe autobuze care merg intre aceleasi orase A si B. Toate autobuzele merg pe drumul de lungime minima intre sursa si destinatie.

Avand date toate rutele autobuzelor in formatul prezentat anterior, trebuie sa determinati ruta <nume_sursa nume_oras_intermediar_1 nume_oras_intermediar_2 ... nume_oras_intermediar_k nume_destinatie> care este frecventata de cei mai multi calatori. Daca vor exista mai multe astfel de rute de aceeasi lungime minima, o veti afisa pe cea mai mica din punct de vedere lexicografic. Se definește ca cel mai frecventat traseu, traseul parcurs de cei mai multi calatori, indiferent de numarul de autobuze care circula pe acel traseu.

Mai multe detalii despre formatul de afisare vor fi prezentate in sectiunea urmatoare.



Atenție! Structurile de date trebuie sa fie implementate generic, nerespectarea acestei cerințe vă va aduce 0p pe tema.

Table of Contents

- Tema 2 - Hit the Road
 - Obiective
 - Intro
 - Enunt
 - Clarificari
 - Generale
 - TASK 1
 - TASK 2
 - TASK 3
 - TASK 4
 - TASK 5
 - Exemplu de functionare
 - Date de intrare
 - Exemplu
 - Date de iesire
 - Checker
 - Exemplu
 - Reguli pentru trimitere
 - Punctaj
 - FAQ

Exemplu de functionare

Vom prezenta in continuare un exemplu de harta rutiera si formatul de intrare si afisare a datelor.

Date de intrare

Ordinea datelor de intrare este urmatoarea:

1. Numarul de orase(N) si numarul de autostrazi(M) din harta rutiera;
2. Lista oraselor sortate lexicografic dupa nume: <nume_0 nume_1 ... nume_N-1>, unde nume_i < nume_i+1 din punct de vedere lexicografic;
3. Urmeaza M linii de forma: <nume_oras_a nume_oras_b>, cu semnificatia ca exista o autostrada unidirectionala de la nume_oras_a la nume_oras_b;
4. Un numar B care reprezinta numarul de benzinarii din harta rutiera, iar pe urmatoarea linie o lista de B orase, de forma: <nume_oras_benzinarie_0 nume_oras_benzinarie_0 ... nume_oras_benzinarie_B-1>, unde nume_oras_benzinarie_i reprezinta un oras de pe harta in care se afla o benzinarie
5. Un numar K, iar pe urmatoarea linie o lista de K nume de orase de forma: <nume_oras_task3_0 nume_oras_task3_1 ... nume_oras_task3_K-1>, unde nume_oras_task3_i reprezinta numele unui oras de pe harta. Orasele din aceasta lista pot fi date in orice ordine
6. Un numar L, urmat de L linii care vor descrie fiecare masina, avand urmatorul format: <nume_oras cantitate_combustibil consum_per_unitate>
7. Un numar P, urmat de P linii care vor descrie fiecare traseu, avand urmatorul format: <nume_sursa nume_destinatie numar_calatori> (explicatia se afla in sectiunea de clarificari)

Exemplu

```
7 8
Boston Chicago LasVegas LosAngeles NewYork Washington Wisconsin
NewYork LasVegas
LosAngeles NewYork
Washington NewYork
LasVegas Washington
Washington LosAngeles
Boston Chicago
Chicago Wisconsin
Wisconsin Boston
2
LasVegas Chicago
3
Washington LosAngeles Wisconsin
2
LosAngeles 5 3
Boston 3 3
4
LasVegas LosAngeles 35
Washington NewYork 40
Boston Wisconsin 25
LasVegas LosAngeles 6
```

Date de iesire

Obs.: Pentru a puncta fiecare task in parte va trebui sa creati un fisier separat pentru afisarea rezultatelor fiecarui task. (Veti crea 5 fisiere task1.out, task2.out etc.)

Format task1.out: nume_oras grad_aglomeratie

Format task2.out: Daca reteaua este valida: HARTA VALIDA
Daca reteaua este invalida: HARTA INVALIDA

Format task3.out: Fisierul va contine K linii de forma Orase inaccesibile pentru
<nume_oras_task3_i>: <nume_oras_inaccesibil_0> <nume_oras_inaccesibil_1>
... <nume_oras_inaccesibil_j>

Format task4.out: Fisierul va contine L linii de forma, fiecare linie i corespunde masinii i din fisierul de intrare: Daca se poate ajunge la o benzinarie: DA -
<combustibil ramas> Daca nu se poate ajunge la o benzinarie: NU

Format task5.out: <numar_total_calatori> <nume_sursa

nume_oras_intermediar_1 nume_oras_intermediar_2 ...
nume_oras_intermediar_k nume_destinatie>

Checker

 Resurse Tema2

Exemplu

task1.out:

NewYork 2

Explicatie: Sunt 2 autostrazi care intra in NewYork (LosAngeles - NewYork si Washington - NewYork)

task2.out:

HARTA INVALIDA

Explicatie: Se observa ca exista mai multe orase intre care nu exista drum (direct sau indirect). Un astfel de exemplu este perechea de orase LosAngeles Boston.

task3.out:

Orase inaccesibile pentru Washington: Boston Chicago Wisconsin
Orase inaccesibile pentru LosAngeles: Boston Chicago Wisconsin
Orase inaccesibile pentru Wisconsin: LasVegas LosAngeles NewYork Washington



task4.out:

NU
DA 0


Explicatie: Prima masina are 5 litri de combustibil initial, un consum de 3 litri per unitate si se afla in LosAngeles. Cea mai apropiata benzinarie de ea este cea din LasVegas. Masina pleaca din LosAngeles si ajunge in NewYork cu 2 litri de combustibil ramasi, deci nu mai poate parcurge nicio alta autostrada si ramane blocata in NewYork. A doua masina are 3 litri de combustibil initial, un consum de 3 litri per unitate si se afla in Boston. Cea mai apropiata benzinarie de ea este cea din Chicago, la care poate ajunge parcurgand muchia Boston - Chicago, ramanand cu 0 litri de combustibil.

task5.out:

41 LasVegas Washington LosAngeles

Explicatie: Pe ruta LasVegas - LosAngeles circula 2 autobuze cu un numar total de 41 de calatori. Drumul de lungime minima de la LasVegas la LosAngeles este LasVegas → Washington → LosAngeles.

Reguli pentru trimitere

Temele vor trebui trimise pe  **vmchecker**. **Atenție!** Temele trebuie trimise în secțiunea **Structuri de Date (CA)**.

Arhiva trebuie să conțină:

- sursele voastre
- fișier Makefile care să conțină obligatoriu următoarele 2 reguli:
 - regula **build**: în urma căreia se generează un executabil numit **tema2**
 - regula **clean** care șterge executabilul (și eventualele fișiere obiect)
- fișier **README** care să conțină detalii despre implementarea temei

Punctaj

1. **80p** teste
2. **Fiecare** test este verificat cu valgrind. Dacă un test are memory leaks, nu va fi punctat.
3. **10p** README + comentarii/claritate cod (**ATENȚIE!** Fișierul README trebuie

făcut explicit, cât să se înțeleagă ce ați făcut în sursă, dar fără comentarii inutile și detalii inutile)

4. **10p** pentru coding-style, proportional cu punctajul obtinut pe teste. De exemplu, pentru o temă care obține maxim pe teste, se pot obține **10p** dacă nu aveți erori de coding style. Pentru o temă care obține 40p pe teste, se pot obține 5p dacă nu aveți erori de coding style.
5. O temă care obține **0p** pe vmchecker este punctată cu **0p**.
6. O tema care nu compilează va fi punctată cu **0p**

Nu copiați!



Toate soluțiile vor fi verificate folosind o unealtă de detectare a plagiatului. În cazul detectării unui astfel de caz, atât plagiatorul cât și autorul original (nu contează cine este sursa) vor primi punctaj **0** pe **toate temele!**

De aceea, vă sfătuim să nu vă lăsați rezolvări ale temelor pe calculatoare partajate (la laborator etc), pe mail/liste de discuții/grupuri etc.

FAQ

Q: Se poate folosi STL?

A: Se pot folosi urmatoarele structuri de date din STL: string, list, vector.

sd-ca/2018/teme/tema2.txt · Last modified: 2019/02/01 13:35 by teodora.serbanescu

 Old revisions

 Media Manager  Back to top

 BY-SA

CHIMERIC DE

W3C CSS

 DOKUWIKI

 GET FIREFOX

 RSS XML FEED

 W3C XHTML 1.0