Proiectarea Algoritmilor -
Tema 2
Dată publicare: 25.04.2015
Deadline soft: 16.05.2015

Ultima modificare: 15.05.2016
Prelungire deadline cu o zi.

1. Problema 1 - **KimLandia**

1.1 Enunț

După o confruntare eroică, Kim a fost ales democratic președintele unei republici prospere, KimLandia. Natural, prima sa prioritate este să construiască o rețea de buncăre, pentru a veni în întâmpinarea unui atac al vecinilor imperialiști.
N buncăre au fost deja construite. Kim trebuie să aleagă între **M** posibile rute **bidirecționale** pentru a le conecta. Un plan de conectare este alcătuit dintr-un subset minim de astfel de legături, astfel încât din fiecare locație să se poată ajunge în orice altă locație. Costul unui plan este dat de suma distanțelor legăturilor.
Deoarece rutele sunt vulnerabile la bombardamente aeriene, Kim vă cere să estimați un plan de **cost minim**. In plus, Kim vă cere să calculați **Q** soluții de rezerva, pentru situația în care una dintre cele **M** legături este sigur inclusă in plan.

1.2 Date de intrare

- Datele de intrare vor fi citite din fișierul **kim.in**.
- Pe prima linie se află 2 numere separate prin spațiu: **N** (numărul de buncăre), **M** (numărul de rute posibile), **Q** (numărul de soluții de rezervă).
- Pe următoarele **M** linii se află triplete de numere (**nod1**, **nod2**, **dist12**), ce descriu o posibilă legatură intre nodurile **nod1** și **nod2**, cu distanța **dist12**.
- Pe următoarele **Q** linii se află câte un număr **Qi**, reprezentând indexul unei muchii care trebuie inclusă într-un plan de rezervă.

1.3 Date de ieșire

- Datele de ieșire vor fi afișate in fișierul **kim.out**.
- Pe prima linie se află un număr reprezentând costul celui mai bun plan, pentru situația in care pot fi folosite oricare dintre cele **M** rute.
- Pe următoarele **Q** linii se află un număr **Ci** reprezentând costul minim al unui plan care conține muchia **Qi**.

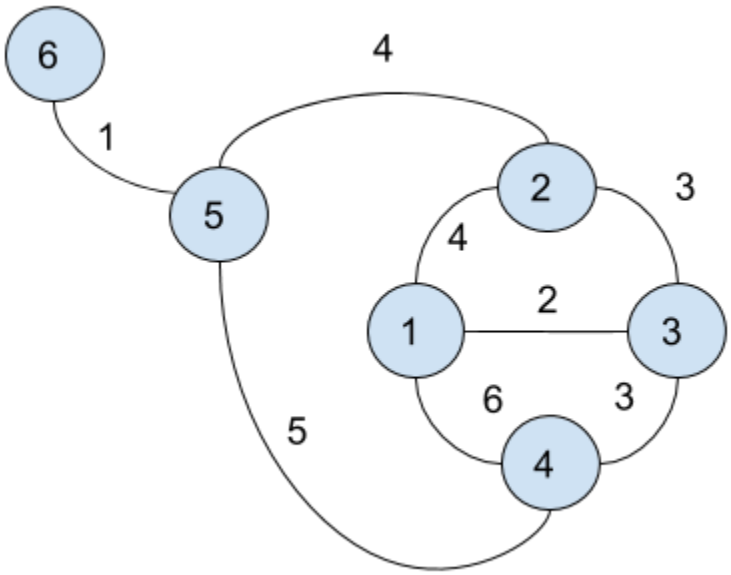
1.4 Precizări

- 1 <= **N** <= 200.000

- **N**-1 <= **M** <= 200.000
- 1 <= **Q** <= 300.000
- Distanța asociată unei rute este o valoare intreagă din intervalul [1, 1.000.000.000]

1.5 Exemplu

kim.in	**kim.out**
6 8 8 1 2 4 4 1 6 3 1 2 2 3 3 5 2 4 3 4 3 4 5 5 5 6 1 1 2 3 4 5 6 7 8	13 14 16 13 13 13 13 14 13



1.6 Limite de timp

C++ 1.6s, Java 4.5s

2. Problema 2 - **Portal**

2.1 Enunț

Știind că rețeaua de drumuri construită anterior poate fi compromisă, Kim a decis să aplice măsuri de protecție suplimentare. Din fericire, în Kimlandia au fost inventate portaluri, o tehnologie inovativă care permite crearea unei legături **unidirecționale** între 2 buncăre.
Un portal este alcătuit din 2 componente care se instalează în buncărele origine și destinație. El permite teleportarea instantanee din origine la destinație, dar doar dacă acestea sunt suficient de apropiate geografic.
Știind perechile de buncăre în care pot fi instalate portaluri, Kim dorește să le amplaseze într-un mod cât mai eficient, astfel incât numărul de buncăre izolate să fie minim. Un buncăr este izolat dacă în acesta nu se poate ajunge din nici un alt buncăr.

2.2 Date de intrare

- Datele de intrare vor fi citite din fișierul `"portal.in"`
- Pe prima linie vor fi 2 numere **N** și **M**, separate prin spațiu, reprezentând numărul de buncăre și portaluri disponibile, respectiv.
- Pe următoarele M linii se vor găsi câte 2 numere, **x** și **y**, separate prin spațiu, cu semnificația că buncărele **x** și **y** sunt suficient de apropiate încât să se instaleze un portal care să le conecteze.

2.3 Date de ieșire

- Rezultatele vor fi scrise în fișierul `"portal.out"`
- Fișierul de ieșire trebuie să conțină o singură linie cu numărul minim de buncăre izolate obținute după instalarea portalurilor.

2.4 Precizări

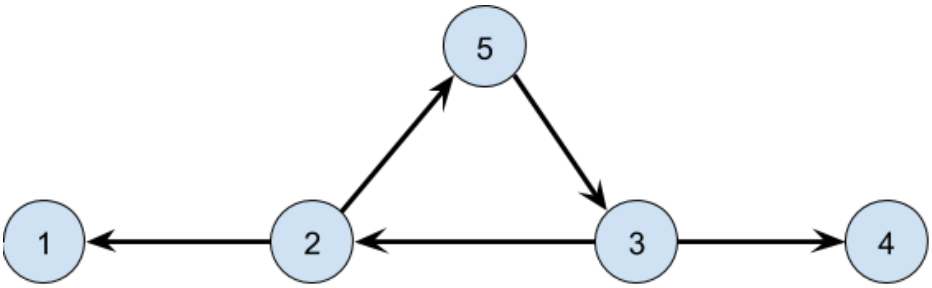
- 2 <= **N** <= 100000
- 1 <= **M** <= 100000

2.5 Exemplu

portal.in	**portal.out**
5 5	0

1 2 2 3 3 4 5 2 5 3	

Mai jos este reprezentat un posibil mod de instalare a portalurilor. Se observă că nici un buncăr nu a rămas izolat.



1.6 Limite de timp

C++ 0.2s, Java 1.4s

Format arhivă și testare

Temele vor fi testate automat pe vmchecker - acesta suportă temele rezolvate in 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 sa fie format zip, cu extensia **.zip** și să conțină în rădăcina acesteia:

- 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 executabilul generat

Atentie! Numele regulilor trebuie să fie exact cele de mai sus, în special cea de run. Absența sau denumirea diferită a acestora va avea drept consecință obținerea a 0 puncte pe testele echivalente problemei rezolvate de regula respectivă.