

Proiectarea Algoritmilor

Tema 3

Data publicare: 6.05.2015

Deadline soft: 22.05.2015

Deadline hard: 27.05.2015

Claudia Cârdei
Clementin Cercel
Vladimir Cernov

Problema 1 - Negustor

Enunț

Gigel a decis să nu mai fie asistent la PA și să devină negustor. Pentru început, el a hotărât să viziteze cele mai mari orașe din țară. Fiind vorba de prima călătorie, Gigel dorește să obțină un profit cât mai bun.

Gigel ia în considerare N orașe din țară, pentru care știe dinainte ce câștig poate obține în fiecare, cât și costul deplasării dintre ele. Drumurile dintre orașele din țara lui Gigel au sens unic de deplasare.

Cerință

Ajutați-l pe Gigel să găsească cel mai bun raport profit/cost, care poate fi obținut efectuând un tur prin orașe (acesta nu trebuie neapărat să conțină toate orașele).

Date de intrare

Fișierul de intrare "negustori.in" va conține pe prima linie două numere N și M ce reprezintă numărul de orașe și respectiv numărul de legături directe între orașe. Următoarea linie va conține N numere ce reprezintă profitul pe care negustorul poate să îl obțină în fiecare oraș. Următoarele M linii conțin câte un triplet (i, j, c) cu următoarea semnificație: este posibilă deplasarea din orașul i în orașul j cu costul c .

Date de ieșire

Fișierul de ieșire "negustori.out" va conține un singur număr (cu precizie de 10^{-5}) ce reprezintă raportul maxim profit/cost.

Exemplu

```
negustori.in
6 9
20 10 30 40 20 100
0 1 2
1 3 3
3 0 4
2 1 1
1 4 2
4 2 3
3 4 1
4 5 1
5 3 11
```

negustori.out
12.5

Explicație: Avem turul 3 -> 4 -> 2 -> 1 -> 3 pentru care raportul este 100 / 8.

Restricții

- $1 \leq N \leq 1000$
- $1 \leq M \leq 10000$
- $p(j) / c(i, j) \leq 1000$, unde $p(j)$ este profitul în orasul j , iar $c(i, j)$ este costul legăturii (i, j)
- Timp execuție pe test: 1,5 secunde C++, 3 secunde Java.

Problema 2 - Prieteni

Enunț

Plictisit, Gigel a hotărât să facă o analiză a grupului de N prieteni din care face parte. El a introdus noțiunea de “prieteni de ordin K ” în felul următor:

- 2 prieteni direcți sunt prieteni de ordin 1
- Dacă 2 persoane sunt prieteni de ordin K , atunci ele sunt și prieteni de ordin $K + 1$
- 2 persoane A și B sunt prieteni de ordinul K dacă există o persoană C a. î. A și C sunt prieteni de ordinul $K - 1$, iar C și B sunt prieteni de ordinul 1.

Cu alte cuvinte, Gigel este prieten de ordinul 1 cu oricare dintre prietenii săi direcți, iar cu prietenii prietenilor săi se află în relație de prietenie de ordinul 2 ș. a. m. d.

Bazându-se pe definiția de mai sus, Gigel a formulat și noțiunea de distanță dintre 2 persoane A , B din grup, fiind egală cu numărul minim de persoane ce trebuie eliminate din grup a. î. A și B să nu fie prieteni de ordin 3.

Cerință

Ajutați-l pe Gigel să calculeze distanța dintre 2 persoane **A** și **B** din grupul său de prieteni.

Date de intrare

Fișierul de intrare “prietenii.in” va conține pe prima linie numărul N . Următoarele N linii vor conține câte N numere, din mulțimea {“0”, “1”}. Al j -lea număr de pe linia i este 1 doar dacă persoanele $i - 2$ și $j - 1$ sunt prieteni de ordin 1. Linia $N + 2$ conține cele 2 numere **A** și **B**.

Date de ieșire

Fișierul de ieșire "prietenii.out" va conține pe o singură linie distanța dintre persoanele **A** și **B**.

Exemplu

```
prietenii.in
```

```
4
```

```
0 1 0 0
```

```
1 0 1 0
```

```
0 1 0 1
```

```
0 0 1 0
```

```
0 3
```

```
prietenii.out
```

```
1
```

Explicație: Trebuie eliminată persoana 1 sau 2.

Restricții

- $2 \leq N \leq 40$
- Timp execuție pe test: 0.5 secunde C++, 1 secundă Java.

Punctare

Punctajul pentru fiecare problemă este de 60 de puncte, obținute prin rularea soluției peste un număr de teste. Testele sunt independente. Fiecare soluție trebuie să se încadreze în limitele de timp menționate.

În afară de rezolvările problemei, se vor acorda 10 puncte pentru coding style și 10 puncte pentru Readme. Un Readme bun trebuie să conțină explicațiile relevante aferente soluțiilor, cât și complexitățile algoritmilor propuși. De asemenea, urmăriți și indicațiile din regulament privind realizarea temelor.

Punctajul total pentru tema 3 este de 140 de puncte.

Format arhivă și testare

Soluțiile vor fi testate automat pe vmchecker, pe care puteți să-l folosiți și voi pentru a verifica corectitudinea algoritmilor. Tema poate fi rezolvată în limbajele de programare

C/C++, Java, pentru folosirea unui alt limbaj trebuie să aveți acordul lui Traian Rebedea.

Arhiva trimisă trebuie să fie în format *.zip și să conțină:

- Fișierele sursă
- Makefile
- Readme

Fișierul Makefile trebuie să conțină următoarele reguli:

- build
- run-p1
- run-p2
- clean

Pentru compilarea surselor C/C++ nu este permisă folosirea opțiunilor de optimizare (-O1, -O2 etc.).

Regulile de mai sus sunt obligatorii, abaterea de la ele poate duce la anularea punctajului (parțial sau total) pentru temă.