

Datatähti 2003 -loppukilpailu

algoritmitehtävä

Tietojenkäsittelytieteen laitos
HELSINGIN YLIOPISTO

6. helmikuuta 2003

1 Afrikan halki -peli

Tätä yksinpeliä pelataan Afrikan tähti -peliklassikon laudalla, mutta eri säännöin:

Pelin lähtökaupungit ovat Kairo ja Tanger mantereen pohjoisosassa, kuten Afrikan tähdesäkin. Pelin kohdekaupunkina on Kapkaupunki mantereen eteläkärjessä. Pelaajan tavoitteena on matkustaa valitsemastaan lähtökaupungista mahdollisimman varmasti kohdekaupunkiin.

Pelin aluksi pelaaja asettaa pelinappulansa valitsemaansa lähtökaupunkiin.

Matkaa taitetaan seuraavasti. Pelaaja valitsee ensin jonkin niistä maan- tai meriteistä, jotka lähtevät siitä kaupungista, jossa hänen pelinappulansa nyt on. (Lentoreittejä ei siis käytetä.) Sitten hän heittää yhtä noppaa. Jos hän saa vähintään niin suuren silmäluvun, että sillä pääsisi Afrikan tähdessä siihen naapurikaupunkiin, johon valittu tie vie, niin hän saa siirtää nappulansa kyseiseen naapurikaupunkiin, ja jatkaa peliä sieltä eteenpäin. Muuten pelaajan maastoauto hajosi valitun tien varrelle, ja peli päättyy siihen — kohdekaupunkiin pääsy jäi haaveeksi.

Olkoon pelaaja esimerkiksi lähdössä Kairosta. Hän valitsee ensimmäiseksi kohteekseen Tripolin, jonne päästäkseen hänen on heitettävä ainakin 4. Hän heittää 5, joten hän saa siirtää pelinappulansa Tripoliin. Sitten pelaajan on aika valita seuraava kohde Tripolista käsin.

Mutta kumpi lähtökaupunki pelaajan kannattaa valita, jotta kohdekaupunkiin pääsy olisi mahdollisimman varmaa?

2 Tehtävä

Toteuta seuraavanlainen ohjelma: Se saa syötteenään pelilaudan, eli kaupungit, kaupunkien väliset tiet, ja näiden teiden vaatimat silmäluvut. Se antaa tulostenaan, kumpi lähtökaupunki kannattaa valita.

3 Syöttö

Ohjelmasi lukekoon syötteensä tekstitiedostosta nimeltä `lauta.in` (tiedoston nimi kirjoitettakoon siis pienin kirjaimin).

```

32
45
1 4 5
1 7 4
1 9 4
1 10 4
2 4 3
2 5 3
2 6 3
3 27 4
4 7 2
5 12 5
6 8 4
7 9 6
9 15 3
10 11 5
10 15 4
10 17 3
11 17 3
11 26 6
12 18 3
13 18 5
13 20 5
14 15 4
14 21 6
15 16 2
15 21 6
16 22 2
17 22 3
18 19 4
19 20 4
20 21 5
20 24 5
21 24 3
22 25 4
22 26 5
22 29 6
24 25 4
24 27 5
26 29 3
27 3 4
27 28 4
28 29 5
28 30 3
29 30 5
29 31 3
31 32 4

```

Kuva 1: Esimerkki syötetiedoston `lauta.in` sisällöstä.

Ensimmäisellä rivillä on kaupunkien lukumäärä $3 \leq M \leq 10\,000$. Toisella rivillä 2 on teiden lukumäärä $0 \leq N \leq 100\,000$. Kolmannesta rivistä alkaen luetellaan nämä N tietä, kukin tie omalla rivillään.

Yksi rivi koostuu kolmesta luvusta p , q ja r , joiden välissä on välilyönti. Ensimmäinen ja toinen luku $1 \leq p < q \leq M$ kertovat ne kaupungit, joiden välillä tämä tie kulkee. Kolmas luku $1 < r \leq 6$ kertoo sen silmäluvun, joka on vähintään heitettävä, jotta tämän tien kulkeminen onnistuu.

Kuvassa 1 on esimerkki syötetiedostosta.

4 Tulostus

Merkintä $p \xrightarrow{r} q$ tarkoittakoon sitä, että kaupungista p pääsee yhtä tietä pitkin kaupunkiin q heittämällä vähintään r . Merkintä

$$\text{Heitto}(r) = \frac{7-r}{6}$$

tarkoittaa puolestaan todennäköisyyttä heittää vähintään tuo r .

Näillä merkinnöillä monesta peräkkäisestä tiestä koostuvan reitin

$$p_0 \xrightarrow{r_1} p_1 \xrightarrow{r_2} p_2 \xrightarrow{r_3} \cdots \xrightarrow{r_s} p_s$$

todennäköisyys on vastaavien onnistuneiden heittojen todennäköisyyksien tulo

$$\text{Heitto}(r_1) \cdot \text{Heitto}(r_2) \cdot \text{Heitto}(r_3) \cdot \cdots \cdot \text{Heitto}(r_s).$$

Nyt ohjelmaltasi vaadittu tulostus voidaan määritellä seuraavasti:

- Jos kummastakaan lähtökaupungista 1 tai 2 ei ole mitään reittiä kohdekaupunkiin 3, niin tulosta luku 3. Se ei esitä mitään kaupunkia, vaan kertoo, ettei koko peliin kannata ryhtyä.
- Jos vain toisesta lähtökaupungista on jokin reitti kohdekaupunkiin, mutta toisesta ei, niin tulosta sen lähtökaupungin numero, josta on reitti.
- Jos molemmista lähtökaupungeista on jokin reitti kohdekaupunkiin, niin vertaa todennäköisintä reittiä lähtökaupungista 1 todennäköisimpään reittiin lähtökaupungista 2. Tulosta sen kaupungin numero, jonka reitti on toista todennäköisempi.
- Jos nämä kaksi reittiä ovatkin yhtä todennäköiset, niin tulosta luku 3. Se kertoo, että voi valita kumman tahansa lähtökaupungin.

Ohjelasi kirjoittakoon tämän tulostuksensa tekstitiedostoon nimeltä `lauta.out` (jonka nimi kirjoitettakoon siis jälleen pienin kirjaimin).

Esimerkiksi kuvan 1 syötetiedostoa vastaavan tulostiedoston `lauta.out` sisällön pitäisi olla

1

koska kohdekaupunkiin 3 pääsee lähtökaupungista 1 todennäköisyydellä

$$\frac{25}{2\,592} \approx 0.009645061728395061$$

kun taas lähtökaupungista 2 todennäköisyys on vain

$$\frac{125}{46\,656} \approx 0.0026791838134430724.$$

5 Pisteytys

Ohjelmaasi testataan 10 syötteellä. Kunkin testin tulos luokitellaan seuraavasti:

Virheetön testitulos antaa testisyötteelle oikean ratkaisun alla selitettävien resurssirajojen sisällä pysyen.

Sellaisesta saat täydet 100 pistettä.

Algoritmitehtävän maksimipistemäärä on siis 10×100 pistettä = 1 000 pistettä.

Kohtalokas virhe on, jos ohjelmasi

- päättyy ajonaikaiseen virheeseen
- päättyy normaalisti mutta tulostaa jotakin muuta kuin sallitun vastauksen 0, 1, 2 tai 3
- tai ylittää seuraavat resurssirajat:

Muistia saa käyttää enintään 128 megatavua.

Tarkemmin sanoen, kun ohjelmaasi suoritetaan Linux-käyttöjärjestelmässä, ei kyseisen prosessin koko saa kasvaa yli tämän rajan. Tähän kokoon lasketaan kaikki ohjelman käyttämä muisti: ohjelmakoodi, aliohjelmakutsupino, dynaamisesti varattu muisti ja käyttämiesi valmiiden aliohjelmakirjastojen kuten `stdio` tarvitsema muisti.

Aikaa saa käyttää enintään 1 sekuntia kun $N \leq 50\,000$ ja 2 sekuntia muuten¹.

Tarkemmin sanoen, tähän ns. keskusyksikköaikaan lasketaan vain se aika, jonka keskusyksikkö viettää suorittamassa itse kirjoittamaasi ohjelmakoodia tai koodisi kutsumaa valmista kirjastoaliohjelmaa kuten `scanf` joka (muun muassa) konvertoi syödetekstiä luvuiksi.

Siihen ei kuitenkaan lasketa sitä aikaa joka kuluu käyttäjärjestelmässä kun `scanf` hakee syödetekstiä tiedostosta muistiin konvertointia varten.

Keskusyksikköaika on siis lyhyempi kuin ohjelman viemä seinäkelloaika. Se, kuinka paljon lyhyempi, riippuu suorittavan tietokoneen muusta samanaikaisesta kuormituksesta. Kevyesti kuormatulla kilpailukoneella ero ei liene suuri.

(Tarkastaja voi myös kasvattaa näitä aikarajoja, jos ne osoittautuvatkin liian tiukoiksi valtaosalle kilpailijoista.)

Sellaisesta et saa lainkaan pisteitä.

Vakava virhe on, jos ohjelmasi tulkitsee pelilaudan pohjana olevaa Afrikan tiekarttaa väärin, eli

- joko vastaa sellaisen lähtökaupungin josta ei olekaan polkua kohdekaupunkiin
- tai vastaa 0 vaikka jokin reitti lähtökaupungista kohdekaupunkiin olisikin olemassa.

Sellaisesta saat vain 1 pisteen.

Paha virhe on, jos ohjelmasi tulkitsee napanheittotodennäköisyyksiä väärin, eli valitsee väärän lähtökaupungin silloin, kun molemmista lähtökaupungeista on reitti kohdekaupunkiin.

Sellaisesta saat 10 pistettä.

Lopuksi kaikkien testien pisteet lastetaan yhteen, ja se on pistemääräsi algoritmitehtävästä.

¹Nämä rajat on valittu kilpailussa käytetyn tietokonemallin laskentatehon mukaan. Tämän mallin keskusyksikkönä oli 500 MHz Pentium III -mikroprosessori.