
Datatähti 2009 -loppukilpailu

Ohjelmointitehtävä 1/2:

Hyper-hyppijä

Eräs harrastelijafyysikoista ja -insinööreistä koostunut joukko kehitti fysiikan lakeja uhmaavan, hyppimällä liikkuvan kulkuneuvon nimeltä "hyper-hyppijä". Kulkuneuvon mullistavin osa on TuplaajaTM-moottori, jonka ansiosta hyper-hyppijä kykenee ällistyttävän pitkiin hyppyihin. Hyper-hyppijällä on kolme erilaista hyppytapaa: hyppy suoraan eteenpäin (E), hyppy suoraan taaksepäin (T) ja ns. "jäähdytyshyppy" (J).

TuplaajaTM-moottorin tuottaman hypyn pituus on suoraan verrannollinen moottorin energiatilan kanssa. Jos energiatila on x , voi hyper-hyppijä hypätä täsmälleen x metriä eteen- tai taaksepäin. Moottorin erikoisuus on se, että sen energiatila kaksinkertaistuu kunkin eteen- tai taaksepäin suuntautuvan hypyn jälkeen. Aluksi moottorin energiatila on 1, joka vastaa 1 yksikön pituista hyppyä. Jarruttaminen tapahtuu jäähdytyshyppyjen avulla. Jäähdytyshyppy suuntautuu suoraan ylöspäin eikä siis sinänsä siirrä hyper-hyppijää eteen- tai taaksepäin. Jäähdytyshypyn seurauksena TuplaajaTM-moottorin energiatila puolittuu. Moottori sammuu, jos energiatila menee pienemmäksi kuin 1 (eli jos energiatilassa 1 tehdään jäähdytyshyppy).

Hyper-hyppijän prototyyppiä testataan suoralla tiellä, joka jatkuu kumpaankin suuntaan äärettömän pitkään. Tehtävänä on kulkea hyper-hyppijällä kyseisen suoran tien pisteestä A pisteeseen B. Pisteiden välinen etäisyys on n metriä. Koska hyper-hyppijä osaa hypätä eteen- tai taaksepäin vain täsmälleen kunkin hetkisen energiatilan määräämän metrimäärän, pitää sen suorittama hyppyjen ketju suunnitella huolellisesti, jotta päästään kätevästi perille täsmälleen pisteen B kohdalle. Tehtäväsi on auttaa toteuttamaan ohjelma, joka laatii hyper-hyppyrille taloudellisen hyppyketjun, jolla päästään kuljettua pisteestä A pisteeseen B. Kukin hyppy tapahtuu suoran tien sisällä, ja on sallittua käydä tien mielivaltaisessa pisteessä (on esim. sallittua ensin hypätä pisteen B ohi ja palata sitten taaksepäin). Hyppyketjun taloudellisuus määritellään sen sisältämien hyppyjen lukumäärän mukaan. Pisteeseen B saapumisen jälkeen moottori pitää sammuttaa (tehdä niin monta jäähdytyshyppyä, että energiatila menee nolnaan). Kuvatun hyppyketjun pitää sisältää myös moottorin sammuttamiseen vaaditut jäähdytyshypyt.

Esimerkki: Halutaan kulkea 12 metriä. Yksi tämän toteuttava hyppyketju on EJEEEJEJJJJ, joka menee seuraavasti:

Suoritettu hyppy	kuljettu matka (hypyn jälkeen)	energiatila (hypyn jälkeen)
(alkutilanne)	0	1
Eteenpäin (E)	1	2
Jäähdytys (J)	1	1
Eteenpäin (E)	2	2
Eteenpäin (E)	4	4
Eteenpäin (E)	8	8
Jäähdytys (J)	8	4
Eteenpäin (E)	12	8
Jäähdytys (J)	12	4
Jäähdytys (J)	12	2
Jäähdytys (J)	12	1
Jäähdytys (J)	12	0

Syöte

Ohjelmasi lukee syötteenä yhden ASCII-tekstitiedoston, jonka nimi on **hyppiija.in**.

Tiedoston ensimmäisellä ja ainoalla rivillä on yksi kokonaisluku n , joka ilmoittaa pisteiden A ja B välisen etäisyyden metreinä. Rivi päättyy rivinvaihtoon. Luvulle n pätee $1 \leq n \leq 2\,000\,000\,000$.

Tuloste

Ohjelmasi kirjoittaa ohjelmoijien kulkutapojen määrän uuteen ASCII-tiedostoon nimeltä **hyppiija.out**.

Tiedoston ainoalle riville täytyy tulostaa m merkistä koostuva merkkijono, joka kuvaa m hypystä koostuvan hyppyketjun. Hypyt kuvataan aiemman esimerkin mukaisesti: kukin merkki on joko 'E', 'T' tai 'J'. Merkkien välissä ei saa olla välilyöntejä tai muuta ylimääräistä, ja kirjaimien koolla on merkitystä (esim. 'e' \neq 'E'). Rivi päättyy rivinvaihtoon.

Toteutustiedosto ja pisteytys

Ohjelmasi toteutuksen tulee olla yhdessä lähdekooditiedostossa. Tiedoston nimen tulee olla ohjelmointikielestä riippuen **hyppiija.c** tai **hyppiija.cpp**.

Ohjelmaasi testataan 20 kertaa erilaisilla syötteillä. Ohjelmasi saa kutakin testiä varten 0.1 sekuntia aikaa ja 192 megatavua keskusmuistia. Kunkin testin kohdalla oikean etäisyyden pienimmällä mahdollisella hyppymäärällä kulkeva hyppyketju antaa 5 pistettä. Jos kuljettu etäisyys on oikein, mutta hyppyketju ei ole pienin mahdollinen, on tuloksena 1 piste. HUOM! Vastauksessa saa kuitenkin olla korkeintaan 65536 hyppyä. Jos hyppyketju on pidempi, ainoastaan 65536 ensimmäistä hyppyä huomioidaan. Väärä vastaus tai aika- tai muistirajan ylittäminen tuottaa 0 pistettä. Tehtävän maksimi on siis 100 pistettä.

Esimerkki syötetiedostosta sekä sitä vastaavasta tulostiedostosta

Esimerkki 1

hyppiija.in:

2

hyppiija.out:

TJEEJJJ

Datatähti 2009 -loppukilpailu

Ohjelmointitehtävä 2/2: Verkkoon!

Takapajustan sai hiljattain uudistusmielisen hallitsijan. Hän päätti tuoda maansa kertaheitolla 2000-luvulle liittämällä maan kaupungit internet-verkkoon. Aiemmin mikään maan kaupunki ei omannut internet-yhteyttä. Hallitsija osaa laskea kaupunkien sijaintikoordinaattien perusteella, paljonko verkkokaapelin vetäminen minkäkin kaupunkiparin välille tulisi maksamaan. Lisäksi kunkin kaupungin osalta tiedetään, paljonko verkkoyhteyden rakentaminen kyseiseen kaupunkiin satelliitin välityksellä maksaisi.

Tehtävänäsi on auttaa hallitsijaa päättämään, miten maan kaikki kaupungit saisi liitettyä internet-verkkoon siten, että satelliittiyhteyksien ja verkkokaapelointien kustannukset olisivat mahdollisimman pienet. Huom! Maa tulee liittää kansainväliseen internet-verkkoon, joka on saatavissa vain satelliitin välityksellä. Ts. kunkin kaupungin tulee joko

- a) olla suoraan yhteydessä satelliittiin, tai
- b) olla kaupunkien väleille vedettyjä verkkokaapeleita pitkin yhteydessä ainakin yhteen sellaiseen kaupunkiin, johon on rakennettu satelliittiyhteys.

Syöte

Ohjelmasi lukee syötteenä yhden ASCII-tekstitiedoston, jonka nimi on **verkkoon.in**. Tiedostossa on $n + 1$ riviä.

Ensimmäisellä rivillä on yksi kokonaisluku n , joka ilmoittaa kaupunkien lukumäärän. Luvulle n pätee $1 \leq n \leq 13\,000$.

Kukin seuraavista n rivistä kuvaa yhden kaupungin. Kaupungit nimetään numeroiden luvuilla $1, \dots, n$, samassa järjestyksessä, kuin ne on annettu syötteessä. Siten kaupunki i on kuvattu rivillä $1+i$.

Kukin tällainen rivi $1 + i$ sisältää kolme toisistaan yhdellä välilyönnillä eroteltua kokonaislukua, jotka ovat järjestyksessä x_i , y_i ja c_i . Arvot x_i ja y_i kuvaavat kaupungin i sijainnin x - ja y -koordinaatit. Arvo c_i kertoo, kuinka monta rahayksikköä satelliittiyhteyden rakentaminen kaupunkiin i maksaa.

Luvuille x_i ja y_i pätee $0 \leq x_i, y_i < 100\,000$, ja luvulle c_i pätee $1 \leq c_i \leq 100\,000$.

Se, kuinka monta rahayksikköä verkkokaapelin vetäminen kaupunkien a ja b välille maksaa, määräytyy ns. korttelietäisyyden mukaan. Tämä lasketaan kaavalla $|x_a - x_b| + |y_a - y_b|$ eli lasketaan yhteen kyseisten kaupunkien x - ja y -koordinaattien erotusten itseisarvot.

Tuloste

Ohjelmasi kirjoittaa vastauksen uuteen ASCII-tiedostoon nimeltä **verkkoon.out**.

Tiedostossa on $m + 1$ riviä.

Tiedoston ensimmäiselle riville täytyy tulostaa kaksi toisistaan yhdellä välilyönnillä eroteltua kokonaislukua: m ja c . Luku m ilmoittaa, kuinka monta yhteyssegmenttiä ehdotukseesi sisältyy: Kuinka monta kahden kaupungin välistä verkkokaapelointia vedetään sekä kuinka monta satelliittiyhteyttä rakennetaan (m on näiden yhteinen kokonaismäärä). Arvo c kertoo, kuinka monta rahayksikköä yhteyksien muodostaminen yhteensä maksaa.

Seuraavat m riviä kuvaavat muodostettavat yhteydet. Kullakin näistä riveistä on kaksi toisistaan yhdellä välilyönnillä eroteltua kokonaislukua: f ja t . Nämä ilmoittavat muodostetun yhteyden päätekaupungit. Jos kyseessä on satelliittiyhteys, ilmoittaa f sen kaupungin, johon satelliittiyhteys rakennetaan, ja tällöin arvon t paikalle tulee tulostaa arvo 0. Jos kyseessä on kahden kaupungin välille vedetty verkkokaapeli, ei arvojen f ja t keskinäisellä järjestyksellä ole merkitystä (kukin yhteys on joka tapauksessa kaksisuuntainen).

Toteutustiedosto ja pisteytys

Ohjelmasi toteutuksen tulee olla yhdessä lähdekooditiedostossa. Tiedoston nimen tulee olla ohjelmointikielestä riippuen **verkkoon.c** tai **verkkoon.cpp**.

Ohjelmaasi testataan 20 kertaa erilaisilla syötteillä. Ohjelmasi saa kutakin testiä varten 1 sekunnin aikaa ja 192 megatavua keskusmuistia. Kunkin testin kohdalla vastaus, joka liittää kaikki kaupungit internet-verkkoon pienimmällä mahdollisella kustannuksella, tuottaa 5 pistettä. Vastaus, joka liittää kaikki kaupungit verkkoon, mutta maksaa minimikustannuksia enemmän, tuottaa 1 pisteen. Virheellinen vastaus tai aika- tai muistirajan ylittäminen tuottaa 0 pistettä. Tehtävän maksimi on siis 100 pistettä.

Esimerkki syötetiedostosta sekä sitä vastaavasta tulostiedostosta

Esimerkki 1

verkkoon.in:

```
8
1 0 2
0 1 3
2 2 3
4 2 3
0 3 2
5 3 3
1 4 2
3 4 3
```

verkkoon.out:

```
8 17
1 2
4 3
5 0
4 6
7 8
1 0
4 0
7 0
```