

Datatähti 2004 -alkukilpailu

1 Helsinki – Vladivostok All Year Long

Ajat työksesi venäläisten uusrikkaiden ostamia länsiautoja Kotkan satamasta itärajan taakse. Tämänkertainen asiakas odottaa autoaan Vladivostokissa, Tyynen valtameren rannalla.

Välissä on valtava Siperia, jossa taajamien väliset matkat ovat pitkiä, eikä joka taajamasta edes löydy bensa-asemaa, jossa voisit auton tankata. Sinun pitää siis suunnitella ajoreittisi siten, että kahden peräkkäisen huoltoaseman välisen matkan pystyy ajamaan yhdellä tankillisella. Asiakas on kärsimätön, ja haluaa sinun valitsevan mahdollisimman lyhyen tällaisen reitin.

1.1 Tehtävä

Toteuta sellainen ohjelma, joka saa syötteenään Siperian tiekartan, ja laskee sen perusteella lyhyimmän mahdollisen ajoreitin pituuden.

Tiekartta koostuu

- taajamista
- tiedosta missä taajamissa on bensa-asemia
- taajamien välisistä maanteistä (joita saa kulkea kumpaankin suuntaan — Siperiassa ei ole yksisuuntaisia teitä)
- näiden teiden pituuksista (1 km tarkkuudella).

Autosi on Škoda Octavia Combi Ambient 1.6i. Sen tankkiin mahtuu 60 l bensaa. Se kuluttaa bensaa 7,5 l/100 km.

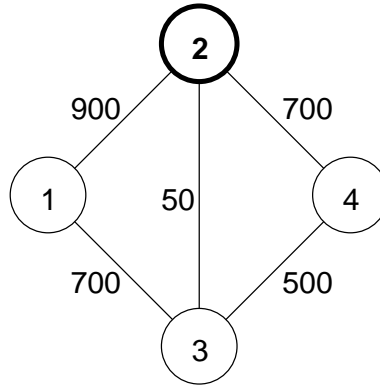
Ohjelmasi laskekoon lyhyimmän sellaisen ajoreitin pituus Kotkasta Vladivostokiin, jota seuraten bensa ei koskaan lopu kesken. Kotkasta lähdetään täydellä tankilla, ja tankki voidaan täyttää vain sellaisissa reitin varrella olevissa taajamissa, joissa on bensa-asema.

1.2 Syöttö

Ohjelmasi lukee syötteensä ASCII-tekstitiedostosta `gasoline.in` seuraavassa muodossa:

- Ensimmäisellä rivillä on seuraavat 3 kokonaislukua välilyönnein eroteltuina tässä järjestyksessä:
 1. Taajamien lukumäärä $2 \leq n \leq 1\,000$. Taajamat on numeroitu $1, 2, 3, \dots, n$. Lähtötaajama (Kotka) on numero 1, kohdettaajama (Vladivostok) on numero n .
 2. Bensa-asemien lukumäärä $0 \leq m \leq n$.
 3. Teiden lukumäärä $0 \leq k \leq 10\,000$.
- Seuraavilla m riveillä annetaan ne taajamat, joissa on bensa-asema. Jokainen taajama annetaan sitä vastaavana numerona omalla rivillään.

Muissa taajamissa ei ole bensa-asemia.
- Lopuilla k riveillä annetaan taajamien väliset maantiet, jokainen tie omalla rivillään. Yhden tien rivi koostuu seuraavista 3 kokonaisluvusta välilyönnein eroteltuina tässä järjestyksessä:



Kuva 1: Esimerkkitiekartta tehtävään 1.

1. Taajama $1 \leq p \leq n$ tien toisessa päässä.
2. Taajamien p ja q välisen tien pituus $1 \leq l \leq 1\,000$ kilometreinä.
3. Taajama $1 \leq q \leq n$ tien toisessa päässä.

Kuvaan 1 on piirretty pieni esimerkki tiekartasta. Ainoa bensa-aseman sisältävä taajama on numero **2**, joka on myös **lihavoitu**. Tätä karttaa vastaava syötetiedosto on seuraava:

```
4 1 4
2
1 900 2
1 700 3
2 50 3
2 700 4
3 500 4
```

1.3 Tulostus

Ohjelmasi tulostaa vastauksensa yhtenä kokonaisukuna ASCII-tekstitiedostoon `gasoline.out` seuraavasti:

- Jos syötteenä saadulla kartalla on lähtötaajamasta 1 kohdetaajamaan n sellainen reitti, jota ajamalla bensa ei lopu kesken, niin lyhyimmän sellaisen reitin kokonaispituus kilometreinä.
- Jos yhtään sellaista reittiä ei ole olemassa, niin 0.

Kuvassa 1 lyhyin tällainen reitti on

$$1 \xrightarrow{700 \text{ km}} 3 \xrightarrow{50 \text{ km}} 2 \text{ (jossa tankkaus)} \xrightarrow{50 \text{ km}} 3 \xrightarrow{500 \text{ km}} 4$$

ja sitä vastaava tulostus on

1300

eli kyseisen reitin yhteispituus $700 + 50 + 50 + 500$.

1.4 Pisteytys

Ohjelmasi ajetaan 10 testisyötteellä. Jokaisessa testissä ohjelmallesi annetaan 64 megatavua työmuistia ja 1 sekunti keskusyksikköaikaa. Jos ohjelmasi tulostaa oikean vastauksen näiden resurssirajojen sisällä, niin sille annetaan tästä testistä 10 pistettä. Muuten sille annetaan tästä testistä 0 pistettä.

Tämä työmuistiraja sisältää kaiken muun ohjelmasi käyttämän muistin paitsi sen suorituskelpoista ohjelmakoodia.

Tämä keskusyksikköaika-rajana sisältää sen ajan, jonka tietokoneen keskusyksikkö käyttää suorittaessaan sinun omaa ohjelmakoodiasi. Se ei sisällä sitä aikaa, jonka se käyttää suorittaessaan käyttöjärjestelmäpalveluita. Erityisesti se ei sisällä sitä aikaa, joka kuluu syötteen lukemiseen tiedostosta `gasoline.in` työmuistiin tai valmiin vastauksen tulostamiseen tiedostoon `gasoline.out`. Toisaalta se sisältää erilaisissa kirjastokutsuissa käytetyn ajan, esimerkiksi luetun syötteen muuntamisen tekstimuodosta vastaavaksi numeroksi.

Ohjelmat testataan sellaisella tietokoneella, jossa keskusyksikkönä on sellainen AMD Athlon -mikroprosessori, jonka kellotaajuus on 1 333 MHz. Sen perusteella voit yrittää arvioida, kuinka pitkä vastaava keskusyksikköaika-rajana on omalla koneellasi. Testikoneen käyttöjärjestelmänä on Linux.

1.5 Lähdekooditiedosto

Kirjoita ratkaisusi yhteen ASCII-lähdekooditiedostoon, jonka nimi riippuu käyttämästäsi ohjelmointikielestä seuraavasti:

KIELI	NIMI
C	<code>gasoline.c</code>
C++	<code>gasoline.cpp</code>
Pascal	<code>gasoline.pas</code>

Laita tiedoston alkuun kommentti, jossa on seuraavat asiat:

- Teksti ”Datatähti 2004 -alkukilpailu”.
- Oma nimesi ja syntymäaikasi.
- Tämän tehtävän nimi ”gasoline”.
- Ilmoita, kääntyykö ohjelmasi kääntäjän oletusvalitsimilla. Jos ei, niin ilmoita lisäksi, mitkä valitsimet on lisättävä kääntäjän komentoriville. Ilmoita vain sellaiset valitsimet, joita ilman ohjelmasi ei käänny!

Tällainen välttämätön lisävalitsin voisi olla esimerkiksi matematiikkakirjastoa käyttävissä C-kielisissä ohjelmissa tarvittava ”-lm”.

Muuten tiedoston kirjoitusasu on vapaa. Älä kuitenkaan käytä yli 80 merkkiä pitkiä rivejä!

Ohjelmointikielillä C ja C++ kirjoitetut lähdekooditiedostot käännetään kääntäjällä GCC <<http://gcc.gnu.org/>>. Ohjelmointikielillä Pascal kirjoitetut tiedostot käännetään puolestaan kääntäjällä Free Pascal <<http://www.freepascal.org/>>, jonka pitäisi olla Turbo Pascal -yhteensopiva. Käytä kuitenkin vain valitsemasi ohjelmointikielen peruspiirteitä yhteensopivuuden takaamiseksi.

Käännökset tehdään kääntäjien oletusarvoisilla valitsimilla, ellei ole ilmoittanut muuta tiedoston alkukommentissa.

Huomioi vielä seuraavat tekniset vaatimukset:

- Ohjelmasi saa lukea syötettä vain tiedostosta `gasoline.in` ja tulostaa vain tiedostoon `gasoline.out`!

Erityisesti se *ei* saa lukea mitään näppäimistöltä tai tulostaa mitään näytölle.

- Kun ohjelmasi suoritus päättyy, niin sen täytyy palauttaa käyttöjärjestelmälle poistumisarvonaan 0.

Ohjelmointikielissä C ja C++ tämä tarkoittaa, että pääohjelman `main` (ja koko ohjelman) suorituksen pitää päättyä komenttoon `return 0;` (eikä vain ”`valua`” viimeiseen sulkevaan kaarisulkuun saakka).

Ohjelmointikielessä Pascal kääntäjä huolehtinee tästä automaattisesti.

Näiden vaatimusten rikkomisesta saattaa seurata pistemenetyksiä testiajojen epäonnistuessa!

A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

Taulukko 1: Taulukko bittilaskutoimituksen XOR arvoille.

Datatähti 2004 -alkukilpailu

2 Kääntöpiiri

Olet töissä elektroniikan sopimussuunnitteluyrityksen laadunvalvontaosastolla. Yritys on suunnitellut asiakkaalleen uuden mikropiirin. Osaston tehtävänä on varmistaa, että tämä piiri toimii kuten pitääkin.

Yksi varmistettava asia on, että piiristä saadaan ulos halutut bitit sopivilla sisään syötetyillä biteillä. Koska piiri on vasta suunniteltu eikä yhtään kappaletta ole vielä valmistettu, täytyy asiasta varmistua sopivalla tietokoneohjelmalla, joka simuloi piirin toimintaa. Tehtävänäsi on kirjoittaa tämä ohjelma.

2.1 Tehtävä

Toteuta sellainen ohjelma, jonka syöte koostuu

- suunnitellun mikropiirin rakenteen kuvauksesta
- halutusta ulos tulevien bittien yhdistelmästä

ja joka kertoo, onko olemassa sellaista sisään menevien bittien yhdistelmää, jolla piiri päästää ulos tämän halutun yhdistelmän.

Piiriin siis menee sisään ja siitä tulee ulos *bittejä* seuraavasti:

- 0 eli ”virta ei kulje”
- 1 eli ”virta kulkee”.

Piiri koostuu erillisistä *porteista*. Yhdellä portilla on kaksi sisäänmenojohtoa ja yksi ulostulojohto. Portin sisäänmenojohtoon on yhdistetty jokin koko piiriin sisään menevistä biteistä. Portin ulostulojohto on yhdistetty johonkin koko piiristä ulos tulevaan bittiin. Portit on siis kytketty *rinnakkain*, ei peräkkäin.

Jokainen portti tekee bittilaskutoimituksen nimeltä XOR. Tämä laskutoimitus on esitetty taulukossa 1. Laskutoimitus ottaa argumentteinaan bitit A ja B , ja antaa vastauksenaan bitin C .

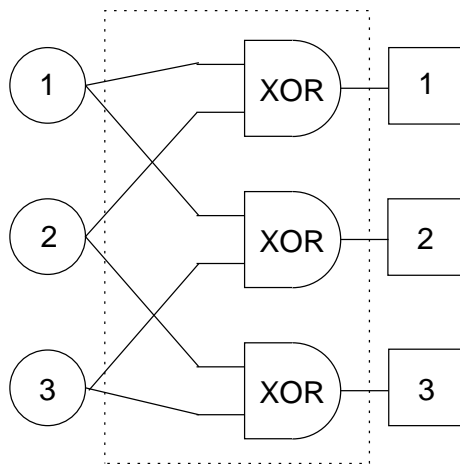
Kuvassa 2 on eräs tällainen piiri. Siinä sisään menevät bitit on merkitty ympyröillä, ja ne on numeroitu 1, 2, 3. Vastaavasti ulos tulevat bitit on merkitty neliöillä, ja nekin on numeroitu 1, 2, 3.

Jos nyt syötteenä saadaan kuvan 2 mikropiiri sekä haluttu ulos tulevien bittien yhdistelmä

$$\text{ulostulo } 1 = 0, \text{ ulostulo } 2 = 0 \text{ ja ulostulo } 3 = 0, \tag{1}$$

niin se voidaan tuottaa asettamalla sisään menevien bittien yhdistelmäksi vaikkapa

$$\text{sisäänmeno } 1 = 1, \text{ sisäänmeno } 2 = 1 \text{ ja sisäänmeno } 3 = 1.$$



Kuva 2: Esimerkki tehtävän 2 mikropiiristä.

Tällä syötteellä ohjelmasi pitää siis vastata ”kyllä”.

Jos taas haluttu ulos tulevien bittien yhdistelmä onkin

$$\text{ulostulo } 1 = \mathbf{1}, \text{ ulostulo } 2 = \mathbf{0} \text{ ja ulostulo } 3 = \mathbf{0}, \quad (2)$$

niin ohjelmasi pitääkin nyt vastata ”ei”: tätä yhdistelmää ei pysty tuottamaan millään sisäänmenojen 1, 2 ja 3 bittiyhdistelmällä.

2.2 Syöttö

Ohjelmasi lukee syötteensä ASCII-tekstitiedostosta `circuit.in` seuraavassa muodossa:

- Ensimmäisellä rivillä on seuraavat 2 välilyönnillä eroteltua kokonaislukua tässä järjestyksessä:
 1. Sisään tulevien bittien lukumäärä $2 \leq n \leq 100\,000$. Nämä bitit on numeroitu $1, 2, 3, \dots, n$.
 2. Ulos lähtevien bittien lukumäärä $1 \leq m \leq 100\,000$. Nämä bitit on numeroitu $1, 2, 3, \dots, m$.
- Lopuilla m rivillä on kunkin ulos lähtevän bitin tiedot. Bitin i tiedot ovat rivinä $i + 1$. Ne on esitetty seuraavilla 3 kokonaisluvulla välilyönnein eroteltuina tässä järjestyksessä:
 1. Yhden sisään tulevan bitin numero $2 \leq p \leq n$.
 2. Toisen sisään tulevan bitin numero $2 \leq q \leq n$.
 3. Ulos lähtevän bitin i haluttu arvo d_i , joko $\mathbf{0}$ tai $\mathbf{1}$.

Tämä rivi tarkoittaa siis, että ulos lähtevän bitin i arvo

- on sisään tulevan bitin numero p ja sisään tulevan bitin numero q arvojen XOR taulukosta 1
- pitää olla d_i .

Kuvissa 3 ja 4 annetaan esimerkkejä syötetiedostoista.

```
3 3
1 2 0
1 3 0
2 3 0
```

Kuva 3: Kuvaa 2 ja yhdistelmää (1) vastaava syötetiedosto.

```
3 3
1 2 1
1 3 0
2 3 0
```

Kuva 4: Kuvaa 2 ja yhdistelmää (2) vastaava syötetiedosto.

2.3 Tulostus

Ohjelmasi tulostaa vastauksensa ASCII-tekstitiedostoon `circuit.out` seuraavasti

- 1 jos on olemassa jokin sellainen sisään tulevien bittien $1, 2, 3, \dots, n$ yhdistelmä, jolla syötteenä saatu mikropiiri tuottaa syötteenä saadun ulos tulevien bittien halutun yhdistelmän

ulostulo 1 = d_1 , ulostulo 2 = d_2 , ulostulo 3 = d_3, \dots , ulostulo $m = d_m$.

- 0 jos yhtään sellaista yhdistelmää ei ole olemassa.

Siis kuvan 3 syötetiedostolla vastauksen pitää olla 1, kun taas kuvan 4 syötetiedostolla vastauksen pitää olla 0.

2.4 Pisteytys

Ohjelmasi ajetaan 10 testisyötteellä. Jokaisessa testissä ohjelmallesi annetaan 64 megatavua työmuistia ja 1 sekunti keskusyksikköaikaa. Jos ohjelmasi tulostaa oikean vastauksen näiden resurssirajojen sisällä, niin sille annetaan tästä testistä 10 pistettä. Muuten sille annetaan tästä testistä 0 pistettä.

Tämä työmuistiraja sisältää kaiken muun ohjelmasi käyttämän muistin paitsi sen suorituskelpoista ohjelmakoodia.

Tämä keskusyksikköaika-rajana sisältää sen ajan, jonka tietokoneen keskusyksikkö käyttää suorittaessaan sinun omaa ohjelmakoodiasi. Se ei sisällä sitä aikaa, jonka se käyttää suorittaessaan käyttöjärjestelmäpalveluita. Erityisesti se ei sisällä sitä aikaa, joka kuluu syötteen lukemiseen tiedostosta `circuit.in` työmuistiin tai valmiin vastauksen tulostamiseen tiedostoon `circuit.out`. Toisaalta se sisältää erilaisissa kirjastokutsuissa käytetyn ajan, esimerkiksi luetun syötteen muuntamisen tekstimuodosta vastaavaksi numeroksi.

Ohjelmat testataan sellaisella tietokoneella, jossa keskusyksikkönä on sellainen AMD Athlon -mikroprosessori, jonka kellotaajuus on 1 333 MHz. Sen perusteella voit yrittää arvioida, kuinka pitkä vastaava keskusyksikköaika-rajana on omalla koneellasi. Testikoneen käyttöjärjestelmänä on Linux.

2.5 Lähdekooditiedosto

Kirjoita ratkaisusi yhteen ASCII-lähdekooditiedostoon, jonka nimi riippuu käyttämästäsi ohjelmointikielestä seuraavasti:

KIELI	NIMI
C	<code>circuit.c</code>
C++	<code>circuit.cpp</code>
Pascal	<code>circuit.pas</code>

Laita tiedoston alkuun kommentti, jossa on seuraavat asiat:

- Teksti ”Datatähti 2004 -alkukilpailu”.
- Oma nimesi ja syntymäaikasi.
- Tämän tehtävän nimi ”`circuit`”.
- Ilmoita, kääntyykö ohjelmasi kääntäjän oletusvalitsimilla. Jos ei, niin ilmoita lisäksi, mitkä valitsimet on lisättävä kääntäjän komentoriville. Ilmoita vain sellaiset valitsimet, joita ilman ohjelmasi ei käänny!

Tällainen välttämätön lisävalitsin voisi olla esimerkiksi matematiikkakirjastoa käyttävissä C-kielisissä ohjelmissa tarvittava ”-lm”.

Muuten tiedoston kirjoitusasu on vapaa. Älä kuitenkaan käytä yli 80 merkkiä pitkiä rivejä!

Ohjelmointikielillä C ja C++ kirjoitetut lähdekooditiedostot käännetään kääntäjällä GCC <<http://gcc.gnu.org/>>. Ohjelmointikielillä Pascal kirjoitetut tiedostot käännetään puolestaan kääntäjällä Free Pascal <<http://www.freepascal.org/>>, jonka pitäisi olla Turbo Pascal -yhteensopiva. Käytä kuitenkin vain valitsemasi ohjelmointikielen peruspiirteitä yhteensopivuuden takaamiseksi.

Käännökset tehdään kääntäjien oletusarvoisilla valitsimilla, ellet ole ilmoittanut muuta tiedoston alkukommentissa.

Huomioi vielä seuraavat tekniset vaatimukset:

- Ohjelmasi saa lukea syötettä vain tiedostosta `circuit.in` ja tulostaa vain tiedostoon `circuit.out`!

Erityisesti se *ei* saa lukea mitään näppäimistöltä tai tulostaa mitään näytölle.

- Kun ohjelmasi suoritus päättyy, niin sen täytyy palauttaa käyttöjärjestelmälle poistumisarvonaan 0.

Ohjelmointikielissä C ja C++ tämä tarkoittaa, että pääohjelman `main` (ja koko ohjelman) suorituksen pitää päättyä komenttoon `return 0;` (eikä vain ”`valua`” viimeiseen sulkevaan kaarisulkuun saakka).

Ohjelmointikielessä Pascal kääntäjä huolehtinee tästä automaattisesti.

Näiden vaatimusten rikkomisesta saattaa seurata pistemenetyksiä testiajojen epäonnistuessa!

3 Ratkaisujen palautus

Tehtävien 1 ja 2 ratkaisujensa palautetaan vastaavat lähdekooditiedostot. Suositeltava tapa on käyttää sähköpostia. Myös kirjepostia voi käyttää, jos on pakko.

Jos sinulla on kysyttävää näistä tehtävistä tai muusta kilpailuun liittyvästä, niin ota yhteyttä sähköpostitse. Myös kirjepostia voi käyttää, jos on pakko, mutta silloin vastauksen saaminen vie kauemmin. Osoitteet annetaan alla.

Kilpailijoiden esittämät kysymykset ja vastaukset niihin tulevat kaikkien kilpailijoiden luettaviksi WWW-sivuston <<http://www.cs.helsinki.fi/matti.nykanen/datatahti/>> kautta. Tällä sivustolla voidaan tiedottaa muustakin kilpailuun liittyvästä, joten seuraa sitä!

Kysymysten pitäisi saapua perille viimeistään torstaina 13. marraskuuta 2003, jotta niihin ehditään vastata hyvissä ajoin ennen kilpailuajan päättymistä.

3.1 Sähköpostilla

Palauta ratkaisusi sähköpostitse osoitteeseen `matti.nykanen@cs.helsinki.fi`. Laadi viesti, jonka otsikko on ”Datatähti 2004 -ratkaisut”, ja joka sisältää nimesi, syntymäaikasi ja yhteystietosi. Liitä sitten jokainen yllä mainituista lähdekooditiedostostasi omana MIME-liitteenään tähän viestiisi.

Sähköpostitse palautetut ratkaisut katsotaan saapuneiksi ajoissa, jos sähköpostin aikaleima on kilpailuajan sisällä. Tämä aikaleima tulee sähköpostin *välittäneestä* palvelinkoneesta, ei omasta mikrotietokoneestasi!

Jos MIME-liitteiden käyttö on mahdotonta, niin lähetä ensin yllä kuvattu viesti, jossa kerrot myös, että lähetät tiedostosi erikseen. Lähetä sen perään kukin lähdekooditiedostostasi omana viestinään. Tässä tapauksessa ratkaisujen katsotaan saapuneen ajoissa, jos *viimeisenkin* viestin aikaleima on kilpailuajan sisällä.

Voit lähettää myös kysymyksiäsi samaan sähköpostiosoitteeseen. Laita silloin viestisi otsikoksi ”Datatähti 2004 -kysymys”. Kysymys ja vastaus ilmestyvät aikanaan yllä mainitulle verkkosivustolle; et siis saa erillistä vastausviestiä.

3.2 Kirjepostilla

Jos et voi käyttää sähköpostia, niin postita sen sijaan ratkaisusi seuraavaan osoitteeseen:

Matti Nykänen
Datatähti 2004 -alkukilpailu
Tietojenkäsittelytieteen laitos
PL 26 (Teollisuuskatu 23)
00014 Helsingin yliopisto

Kirjeitse palautetut ratkaisut katsotaan saapuneiksi ajoissa, jos kirjekuoren postileima on kilpailuajan sisällä.

Kirjekuoren pitää sisältää seuraavat esineet:

- Yksi 3,5 tuuman DOS-formatoitu levyke, jonka juurihakemistoon olet tallettanut kyseiset lähdekooditiedostot.

Talleta lisäksi samaan juurihakemistoon ASCII-tekstitiedosto nimeltä `vastaja.txt` johon olet kirjoittanut oman nimesi, syntymäaikasi ja yhteystietosi sekä tiedon, että kyseessä on osallistuminen Datatähti 2004 -alkukilpailuun.

- Mielellään toinen samanlainen levyke varmuuden vuoksi.
(Ensimmäinen levyke voi kärsiä matkalla!)

- Tiedosto `vastaaja.txt` paperille tulostettuna.

(Siltä varalta että *molemmat* levykkeet kärsivät matkalla...)

Pakkaa levykkeet huolellisesti (käyttäen esimerkiksi pehmustettua kirjekuorta).

Voit lähettää myös kirjallisia kysymyksiäsi tähän samaan osoitteeseen. Liitä kysymyksiisi osoite, johon vastauksen voi lähettää.

(Julkaisupäivä: 04.11.2003.)