
Datatähti 2009 -alkukilpailu

Ohjelmointitehtävä 1/3: Hissimatka

HUOM: Tutustuthan huolellisesti tehtävien sääntöihin ja palautusohjeisiin (sivu 7)

Joukko ohjelmoijia on talon pohjakerroksessa, ja he haluavat päästä ylimpään kerrokseen tietokonehuoneeseen. Talossa on hissi, ja portaitakin voi käyttää. Mutta kuinka monta tapaa on valita, ketkä menevät hissillä ja ketkä jalkaisin?

Kaikki ohjelmoijat lähtevät matkaan yhtä aikaa, ja kaikki hissiä käyttävät kulkevat samassa hississä.

Tarkastellaan esimerkkiä, jossa ohjelmoijia on neljä ja hissin lasti saa painaa korkeintaan 200 kg. Ohjelmoijien painot ovat:

ohjelmoija	paino (kg)
A	70
B	130
C	50
D	75

Nyt ohjelmoijat voivat siirtyä ylimpään kerrokseen 11 tavalla:

hissillä	jalkaisin	paino hississä (kg)
-	A, B, C, D	0
C	A, B, D	50
A	B, C, D	70
D	A, B, C	75
A, C	B, D	120
C, D	A, B	125
B	A, C, D	130
A, D	B, C	145
B, C	A, D	180
A, C, D	B	195
A, B	C, D	200

Muissa tapauksissa hissin lasti olisi liian raskas.

Tehtävänäsi on toteuttaa ohjelma, joka laskee, kuinka monella tavalla ohjelmoijat voivat siirtyä talon ylimpään kerrokseen, kun ohjelmalle annetaan hissin lastin maksimipaino ja ohjelmoijien painot.

Syöte

Ohjelmiasi lukee syötteenä yhden ASCII-tekstitiedoston, jonka nimi on **hissi.in**.

Tiedoston ensimmäisellä rivillä on kokonaisluku m , joka on hissin lastin maksimipaino. Toisella rivillä on kokonaisluku n , joka on ohjelmoijien määrä. Lopuksi tulee n riviä, joilla jokaisella on kokonaisluku p_i : yhden ohjelmoijan paino. Luvuille pätee $1 \leq m \leq 10000$, $1 \leq n \leq 20$ ja $1 \leq p_i \leq 500$.

Tuloste

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

Tiedoston ainoalle riville täytyy tulostaa kokonaisluku t : kuinka monta tapaa on valita, ketkä menevät hissillä ja ketkä jalkaisin?

Toteutustiedosto ja pisteytys

Ohjelmasi toteutuksen tulee olla yhdessä lähdekooditiedostossa. Tiedoston nimen tulee olla ohjelmointikielestä riippuen **hissi.c**, **hissi.cpp** tai **hissi.pas**.

Ohjelmaasi testataan 20 kertaa erilaisilla syötteillä. Ohjelmasi saa kutakin testiä varten 1 sekunnin aikaa ja 192 megatavua keskusmuistia. Kunkin testin kohdalla oikea tuloste antaa 5 pistettä ja väärä 0 pistettä. Aika- tai muistirajan ylittäminen tuottaa tulosteen oikeellisuudesta riippumatta 0 pistettä. Tehtävän maksimi on siis 100 pistettä.

Esimerkkejä syötetiedostoista sekä vastaavista tulostiedostoista

Esimerkki 1

hissi.in:

200

4

70

130

50

75

hissi.out:

11

Esimerkki 2

hissi.in:

329

6

89

73

115

87

156

95

hissi.out:

35

Datatähti 2009 -alkukilpailu

Ohjelmointitehtävä 2/3: Pensaslabryntti

HUOM: Tutustuthan huolellisesti tehtävien sääntöihin ja palautusohjeisiin (sivu 7)

Rouva Kasvinen näki televisiosta ohjelman Wienin Schönbrunnin palatsin puutarhassa olevasta pensaslabryntista. Ohjelman innoittamana hän päätti tehdä pensaslabryntin myös oman talonsa puutarhaan. Rouva Kasviselle sattui kuitenkin pieni erhe: hän kasvatti puutarhaan niin paljon pensasaitaa, ettei labryntissa enää ollutkaan kulkureittiä labryntin alkupisteestä A loppupisteeseen B. Tämän vuoksi puutarhasta pitää poistaa osa pensasaidoista. Kasvien suurena ystävänsä rouva Kasvinen haluaa poistaa niin vähän pensasaitaa kuin mahdollista.

Rouva Kasvisen puutarha voidaan kuvata suorakulmion muotoisena ruudukkona, jossa on m riviä ja n saraketta. Kukin ruutu on joko vapaa tai siinä on pensasaidan pala. Labryntin alkupiste A sijaitsee ruudukon vasemmassa yläkulmassa eli 1. rivin 1. sarakkeessa. Labryntin loppupiste B sijaitsee ruudukon oikeassa alakulmassa eli m . rivin n . sarakkeessa. Labryntissa on mahdollista kulkea vapaasta ruudusta johonkin sen viereiseen vapaaseen ruutuun. Ruudun viereisiksi ruuduiksi lasketaan sen vasemmalla, oikealla ja ylä- tai alapuolella olevat ruudut. Puutarharuudukon ulkopuolelle ei saa kulkea.

Pensasaitaa on mahdollista poistaa pala kerrallaan. Palan poistamisen jälkeen sen sijaintiruutu muuttuu vapaaksi. Tehtävänäsi on toteuttaa ohjelma, joka auttaa rouva Kasvista laskemaan, mikä on pienin mahdollinen määrä pensasaidan paloja, jotka poistamalla on mahdollista aikaansaada vapaista ruuduista koostuva kulkureitti labryntin alkupisteen A ja loppupisteen B välille.

Syöte

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

Tiedostossa on $m + 1$ riviä.

Ensimmäisellä rivillä on kaksi toisistaan yhdellä välilyönnillä eroteltua kokonaislukua m ja n , tässä järjestyksessä. Luku m ilmoittaa puutarharuudukon korkeuden (= rivien lukumäärän) ja luku n leveyden (= sarakkeiden lukumäärän).

Kukin seuraavista m rivistä sisältää n peräkkäistä merkkiä (ilman välilyöntejä), jotka kuvaavat yhden puutarharuudukon rivin. Kukin merkki on joko 'V' tai 'P'. Merkki 'V' tarkoittaa, että vastaava ruutu on vapaa. Merkki 'P' tarkoittaa, että vastaavassa ruudussa on pensasaidan pala. Rivit annetaan järjestyksessä ylhäältä alaspäin, ja yksittäisen rivin kuvaavat merkit järjestyksessä vasemmalta oikealle. Näin ollen tiedoston **pensas.in** rivillä $1 + i$ ja sarakkeessa j oleva merkki vastaa ruudukon i . rivin j . saraketta.

Luvuille pätee $1 \leq m, n \leq 3000$.

Tuloste

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

Tiedostossa on $c + 1$ riviä.

Tiedoston ensimmäiselle riville täytyy tulostaa kokonaisluku c , joka ilmoittaa kuinka monta pensasaidan palaa on vähintään poistettava, jotta labryntin alkupisteen A ja loppupisteen B välille muodostuu vapaa reitti.

Seuraavat *c* riviä ilmoittavat sellaiset ruudut, joiden pensaspalat poistamalla vapaa reitti syntyy. Kukin näistä riveistä sisältää kaksi toisistaan yhdellä välilyönnillä eroteltua kokonaislukua, joista ensimmäinen ilmoittaa ruudun rivin ja toinen ruudun sarakkeen.

Toteutustiedosto ja pisteytys

Ohjelmasi toteutuksen tulee olla yhdessä lähdekooditiedostossa. Tiedoston nimen tulee olla ohjelmointikielestä riippuen **pensas.c**, **pensas.cpp** tai **pensas.pas**.

Ohjelmaasi testataan 20 kertaa erilaisilla syötteillä. Ohjelmasi saa kutakin testiä varten 2 sekuntia aikaa ja 192 megatavua keskusmuistia. Kunkin testin kohdalla täysin oikea tuloste antaa 5 pistettä. Vapaan reitin tuottava, mutta turhan monta pensasaidan palaa poistava vastaus antaa 1 pisteen. Vapaata reittiä tuottamaton vastaus antaa 0 pistettä. Aika- tai muistirajan ylittäminen tuottaa tulosteen oikeellisuudesta riippumatta 0 pistettä. Tehtävän maksimi on siis 100 pistettä.

Esimerkkejä syötetiedostoista sekä vastaavista tulostiedostoista

Esimerkki 1

pensas.in:

6 7

VVVVVVV

PVPPVPP

VVPVVPP

VPPPPPV

VPVVVPP

VVVPVPV

pensas.out:

1

6 6

Esimerkki 2

pensas.in:

10 10

VPVPVPPPP

PPVPPPPPP

PVPPPPPPV

PPPPPPVPP

PPPVPVPPP

VPPPVPVPP

VPPPPVPPP

PPPPVPPPP

VPPVPPVPP

PPVPPPPVV

pensas.out:

7

1 2

3 4

4 4

5 5

6 6

7 8

8 8

Datatähti 2009 -alkukilpailu

Ohjelmointitehtävä 3/3: Professorin algoritmi

HUOM: Tutustuthan huolellisesti tehtävien sääntöihin ja palautusohjeisiin (sivu 7)

Professori Bittinen on kehittänyt nerokkaan algoritmin, joka laskee kokonaisluvun neliöjuuren. Algoritmista on kuitenkin yksi heikkous: se toimii oikein vain silloin, kun luvun binääriesityksessä ei ole vierekkäin kahta ykkösbittää.

Esimerkiksi algoritmi laskee oikein lukujen 8 ja 34 neliöjuuret, koska lukujen binääriesitykset ovat 1000 ja 100010. Vastaavasti algoritmi käsittelee virheellisesti luvut 11 ja 29, koska niiden binääriesitykset ovat 1011 ja 11101.

Professori haluaa arvioida algoritminsä käyttökelpoisuutta tutkimalla tietyn lukuvälin lukuja. Hän haluaa selvittää, kuinka suuri osa luvuista kelpaa algoritmille eli kuinka monen luvun binääriesityksessä ei ole vierekkäin kahta ykkösbittää.

Esimerkiksi lukuvälillä 3–7 lukujen 3, 4, 5, 6 ja 7 binääriesitykset ovat 11, 100, 101, 110 ja 111. Professorin algoritmi selviytyy luvuista 4 ja 5 muttei muista lukuvälin luvuista.

Tehtävänäsi on toteuttaa ohjelma, joka auttaa professoria laskemaan, kuinka monta lukua tietyllä lukuvälillä algoritmi käsittelee oikein.

Syöte

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

Tiedostossa on kaksi riviä, joista ensimmäisellä on kokonaisluku a ja toisella kokonaisluku b . Luku a on ensimmäinen tutkittava luku, ja luku b on viimeinen tutkittava luku. Luvuille pätee $1 \leq a < b < 10^{18}$.

Tuloste

Ohjelmasi kirjoittaa kelvollisten lukujen määrän uuteen ASCII-tiedostoon nimeltä **profalg.out**.

Tiedoston ainoalle riville täytyy tulostaa kokonaisluku c , joka on niiden annetulla lukuvälillä olevien lukujen määrä, joiden binääriesityksessä ei ole vierekkäin kahta ykkösbittää.

Toteutustiedosto ja pisteytys

Ohjelmasi toteutuksen tulee olla yhdessä lähdekooditiedostossa. Tiedoston nimen tulee olla ohjelmointikielestä riippuen **profalg.c**, **profalg.cpp** tai **profalg.pas**.

Ohjelmaasi testataan 20 kertaa erilaisilla syötteillä. Ohjelmasi saa kutakin testiä varten 1 sekunnin aikaa ja 192 megatavua keskusmuistia. Kunkin testin kohdalla oikea tuloste antaa 5 pistettä ja väärä 0 pistettä. Aika- tai muistirajan ylittäminen tuottaa tulosteen oikeellisuudesta riippumatta 0 pistettä. Tehtävän maksimi on siis 100 pistettä.

Esimerkkejä syötetiedostoista sekä vastaavista tulostiedostoista

Esimerkki 1

profalg.in:

3

7

profalg.out:

2

Esimerkki 2

profalg.in:

123

321

profalg.out:

44

Sääntöjä ja ohjeita

Mitä palautetaan?

Kunkin ohjelmointitehtävien vastaus koostuu tehtävän yhteydessä mainitulla tavalla nimitystä yhdestä lähdekooditiedostosta. Siis jokaista tehtävää kohden oma tiedosto. Älä lähetä esimerkiksi käännettä ohjelmatiedostoa.

Sallitut ohjelmointikielet

Kunkin ohjelmointitehtävän vastauksessa sallitut ohjelmointikielet ovat C, C++ ja Pascal.

Lähdekooditiedostot

Lähdekooditiedoston nimen tiedostopäätte määrätty seuraavasti:

kieli	tiedostopäätte
C	c
C++	cpp
Pascal	pas

Kunkin lähdekooditiedoston alussa pitää olla kommenttiosio, joka sisältää seuraavat tiedot:

1. Otsikkoteksti "Datatähti 2009 -alkukilpailu"
2. Tehtävän numero
3. Nimesi
4. Syntymäaikasi
5. Koulusi/lukiosi nimi ja paikkakunta

Ohjelmakoodin kirjoitusasu on muuten vapaa (esim. koodin selkeyttä ei arvostella).

Ohjelman toiminta

Ohjelmasi on noudatettava alla lueteltuja vaatimuksia, tai muuten seurauksena saattaa olla pistemenetyksiä.

- Ohjelma saa lukea ja kirjoittaa vain tehtävässä nimettyjä tiedostoja. Erityisesti näppäimistöltä lukeminen tai näytölle tulostaminen on kiellettyä.
- Tulostiedoston täytyy aina päättyä rivinvaihtoon.
- Ohjelmointikielissä C ja C++ pääohjelman main suorituksen pitää päättyä lauseeseen `return 0;` (Pascal-kääntäjä huolehtii tästä puolestasi).
- Ohjelmointikielessä pitää rajoittua vain sen standardoituihin peruspiirteisiin, ja välttää laite- tai kääntäjäkohtaisia erikoisuuksia.
 - Ohjelmointikielissä C ja C++ on käytössä standardoitu matematiikkakirjasto (eli käännettäessä annetaan valitsin `-lm`).
 - Yleisesti ottaen saa käyttää ohjelmointikielten standardikirjastoja (esim. C++-kielen Standard Template Library (STL), C-kielen peruskirjasto (C standard library), Java-kielessä Java Platform SE:n luokat).

Koodin ajoympäristö

Testitietokoneessa on Linux käyttöjärjestelmä sekä seuraavat kääntäjät:

Kieli	Kääntäjä	Versio	Kielistandardi (valitsimet)	Optimointitaso
C	gcc	≥ 4.0	ANSI/ISO C89/90 (-ansi)	-O2 -static
C++	g++	≥ 4.0	ISO C++ (-ansi)	-O2 -static
Pascal	Free Pascal	≥ 2.0	Borland Turbo Pascal 7.0 (-Mtp -Sg)	-O2 -XS

Resurssirajat

Tehtävien kohdalla todettiin, että ohjelma saa käyttää kunkin testiajon aikana vain tietyn määrän aikaa ja muistia.

Em. resurssirajat asetetaan käyttäen Linuxin `ulimit` komentoa. Esimerkiksi 1 sekunnin aika- ja 192 megatavun muistiraja asetetaan komennolla `'ulimit -t 1 -v 196608'`.

Voit arvioida testikoneen nopeutta sen pohjalta, että siinä on AMD Athlon64 3200+ CPU.

Ohjelmointitehtävien vastausten palautus

Jos samalta henkilöltä tulee kilpailuaikana useita eri ratkaisuja, niin viimeinen niistä arvostellaan.

Jos kilpailija lähettää vastauksensa myöhästyneenä, voidaan vastaus jättää huomiotta.

Ratkaisujen palautus tapahtuu sähköpostitse osoitteeseen heikki.hyyro@cs.uta.fi. Viestin rakenne on seuraava:

- Sähköpostin otsikkokenttään (subject) teksti “Datatähti 2009 -alkukilpailun vastaukset”.
- Viestin rungoksi omat yhteystietosi:
 1. Nimesi
 2. Syntymäaikasi
 3. Koulusi/lukiosi nimi ja paikkakunta
 4. Postiosoitteesi
 5. Puhelinnumerosi
 6. Sähköpostiosoitteesi
- Lähdekooditiedostot laitetaan sähköpostin mukaan erillisessä zip-pakkauksessa. Zip-pakkaus tulee nimetä muodossa **etunimi_sukunimi.zip**.
 - Tämän toiminnon nimi on yleensä “Liitä tiedosto”, “Attach File” tai vastaava.

Sähköpostivastauksen katsotaan lähteneen ajoissa, jos sen aikaleima on kilpailuajan sisällä. Tämä aikaleima katsotaan sähköpostin välittäneestä palvelinkoneesta.

Kysymyksiä (ja vastauksia)

Voit esittää sähköpostitse täsmennyskysymyksiä ohjelmointitehtäviin tai niiden palautukseen liittyen. Käytä edellä annettua vastausten palautusosoitetta. Kysymykset ja vastaukset laitetaan kilpailun aikana näkyviin kisasivustolle ilman kysyjien nimiä. Sivuja kannattaa siis seurata!