

Yleiset ohjeet

Tehtävänä on laatia toimiva ohjelma seuraavilla sivuilla esitetyn algoritmisen ongelman ratkaisemiseksi. Ohjelmointiympäristöinä sinulla on valittavanasi Turbo C++, joka käynnistyy komennolla TC, ja Borland Pascalia, joka käynnistyy komennolla BP. Aikaa ohjelman toteuttamiseen sinulla on kolme tuntia. Lopputuloksena tulee olla suorituskelpoinen ohjelmatiedosto, joka sinun tulee *ennen kilpailuajan päättymistä* kopioida saamallasi levykkeelle numero 1 juurihakemistoon tiedostoksi

SOKKELO.EXE

Toiselle levykkeelle kilpailun järjestäjät ottavat varmuuskopion ajan päätyttyä. Ohjelman tulee lukea syötteen ja tulostaa tulosteet oletushakemiston tiedostoja käyttäen tehtävänkuvauksessa esitetyllä tavalla. *Ohjelman ei pitäisi tulostaa mitään näyttöruudulle.*

Levykkeillä on kopiot tehtävän yhteydessä esitetyistä esimerkkitiedostoista. Saat kilpailun tulosten vahvistamisen jälkeen kopion omasta levykkeestäsi, joten voit kopioida sille myös esim. oman ohjelmasi lähdekoodin tai muuta haluamaasi materiaalia. Muista kuitenkin *kääntää ohjelman viimeinen versio* levykkeelle ennen ajan loppumista. Kiintolevyä käyttäessäsi tallenna kaikki tiedostosi hakemistoon \USER, jotta ne saadaan helposti siivotuksi suoritukseksi päätyttyä.

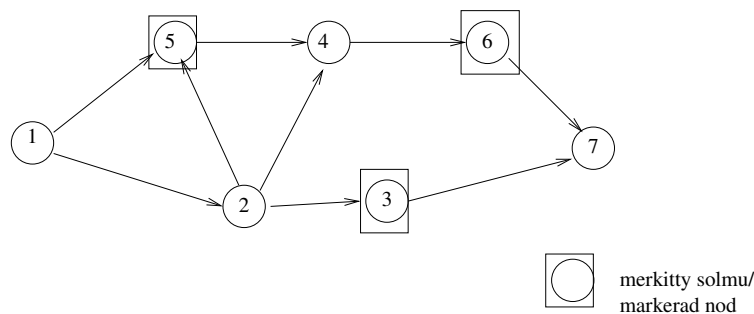
Jos haluat kilpailun kestäessä tulostaa jonkin tiedoston paperille, kopioi se levykkeelle hakemistoon \PRINT ja anna levyke valvojalle. Tiedostot tulostetaan, ja saat takaisin levykkeen ja paperitulosteet. Tämä voi kestää jonkin aikaa.

Sokkelo

Tarkastellaan koejärjestelyä, jossa rotta laitetaan sokkelon sisäänmenoaukkoon ja yrittää selviytyä ulos sokkelon toisesta päästä. Matkalla sillä on lisäksi mahdollisuus syödä sokkeloon sijoitettuja juustonpaloja.

Sokkelo koostuu risteyksistä ja niiden välisistä yhdyskäytävistä. Mihin tahansa risteykseen voidaan sijoittaa pala juustoa. Sokkelon sisäänmeno- ja ulostuloaukko tulkitaan muuten kumpikin risteyksiksi, mutta niissä ei koskaan ole juustoa. Yhdyskäytävät ovat yksisuuntaisia. (Niissä on jokaisessa luukku, jonka rotta saa auki vain toiselta puolelta.) Jotta rotta ei voisi kokonaan eksyä sokkeloon, yhdyskäytävien suunnat on asetettu niin, että rotan on mahdotonta palata takaisin risteykseen, jossa se on jo ollut. Sokkelossa ei myöskään ole umpikujia: sisäänmenoaukosta johtaa ainakin yksi reitti jokaiseen muuhun risteykseen, ja jokaisesta risteyksestä johtaa reitti ulospääsyaukkoon.

Sokkelo voidaan siis mallintaa suunnattuna syklittömänä verkkona, jossa on erityinen alkusolmu ja loppusolmu. Verkon solmut esittävät risteyksiä ja kaaret yhdyskäytäviä. Syklittömyys tarkoittaa, että kaarista ei voi muodostaa "silmukkaa". Osa solmuista on merkitty (siis vastaa juustollista risteystä). Oheisessa kuvassa on esimerkki tällaisesta verkosta.



Tehtävässä tarkastellaan kolmea tällaisiin verkkoihin liittyvää osaongelmaa. Sanotaan alkusolmusta loppusolmuun johtavan polun *pituuiseksi* sillä olevien solmujen lukumäärää (päätösolmut mukaanlukien) ja *arvoksi* sillä olevien merkittyjen solmujen lukumäärää (siis rotan juustosaalista).

1. Halutaan tietää lyhin polku alkusolmusta loppusolmuun piittaamatta siitä, mitkä solmut ovat mahdollisesti merkittyjä. (Tämä siis vastaa rotan optimireittiä, kun se ei ole nälkäinen.)
2. Halutaan tietää arvoltaan suurin mahdollinen polku. (Tämä siis vastaa rotan optimireittiä, kun se on nälkäinen mutta ei kiireinen.)
3. Halutaan tietää, mikä olisi arvokkain mahdollinen polku, jos verkkoon saataisiin lisätä *yksi* uusi kaari. Lisättävän kaaren pitää olla sellainen, että verkossa ei lisäyksenkään jälkeen ole syklejä. Muuten lisättävä kaari valitaan vapaasti siten, että syntyy arvoltaan mahdollisimman suuri polku. (Tämä vastaa siis koejärjestelyn muuttamista lisäämällä sellainen yksi yhdyskäytävä, että rotan mahdollinen juustosaalis kasvaa mahdollisimman paljon. Uudessakaan sokkelossa rotta ei saa voida joutua kokonaan eksyksiin.)

Tehtävänäsi on kirjoittaa ohjelma, joka lukee verkon kuvauksen syöttötiedostosta ja tulostaa kahteen eri tulostiedostoon vastauksen kolmeen edelläesitettyyn kysymykseen. Jos oikeita vastauksia on useita (esim. verkossa on useita yhtä lyhyitä polkuja), voit tulostaa minkä tahansa näistä oikeista vastauksista.

Syöttötiedot

Ohjelmasi tulee lukea syöte oletushakemiston tiedostosta VERKKO.TXT.

Tiedoston ensimmäisellä rivillä on yksi kokonaisluku n , verkon solmujen lukumäärä. Solmujen lukumäärälle pätee $2 \leq n \leq 2000$.

Tiedoston loput rivit jakautuvat n osaan, joista kukin on yhden verkon solmun i kuvaus, $1 \leq i \leq n$. Solmut on numeroitu, ja tiedostossa on ensin solmun 1 kuvaus, sitten solmun 2, ja niin edelleen, aina solmuun n saakka. Alkusolmu on aina numeroltaan 1 ja loppusolmu n .

Solmun i kuvauksessa on ensin yhdellä rivillä kaksi kokonaislukua, k_i ja b_i , joista $0 \leq k_i \leq n-1$ ja $b_i \in \{0, 1\}$. Näistä k_i on solmusta i lähtevien kaarten lukumäärä ja b_i on 1, jos solmu i on merkitty, ja 0 muuten. Seuraavilla k_i rivillä on niiden solmujen numerot, joihin on kaari solmusta i .

Kaikkiaan tiedostossa on siis $1 + n + m$ riviä, missä $m = \sum_{i=1}^n k_i$ on verkon kaarien kokonaislukumäärä. Voit olettaa, että $m \leq 40000$.

Ylläolevan kuvan mukainen esimerkkiverkko esitettäisiin syöttötiedostossa VERKKO.TXT seuraavasti:

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

Tulostustiedot

Ohjelmasi tulee tulostaa vastauksensa oletushakemiston kahteen eri tiedostoon POLUT.TXT ja KAARI.TXT. Tiedostoissa ei saa olla muuta kuin tarkalleen alla kuvatut tiedot, ei ylimääräisiä välilyöntejä, tyhjiä rivejä tms. Tiedoston lopussa pitää kuitenkin olla (tavalliseen tapaan) yksi rivinvaihto.

Tiedostossa POLUT.TXT on kaksi alkusolmusta loppusolmuun vievää polkua. Ensimmäisen polun pitää olla mahdollisimman lyhyt. Toisen polun pitää olla arvoltaan mahdollisimman suuri. Kumpikin polku on esitetty luettelemalla polun solmujen numerot järjestyksessä alkusolmusta loppusolmuun. Jokainen solmun numero on omalla rivillään, ja lisäksi tiedostossa on *yksi* tyhjä rivi erottamassa näitä kahta polkua.

Tiedoston KAARI.TXT ensimmäisellä rivillä on kaksi kokonaislukua i ja j välilyönnillä toisistaan erotettuina. Näiden pitäisi olla sellaiset, että kun alkuperäiseen verkkoon lisätään kaari solmusta i solmuun j , niin uudessa verkossa arvokkaimman polun arvo on mahdollisimman suuri. Seuraavilla riveillä on lueteltuna tämän uuden arvokkaimman polun solmut samalla tavoin kuin polut tiedostossa POLUT.TXT.

Esimerkkitapauksessa eräs oikea tulostiedosto POLUT.TXT siis olisi

1
2
3
7

1
2
5
4
6
7

ja tiedosto KAARI.TXT

6 3
1
5
4
6
3
7

Arvostelu

Ratkaisusi arvostellaan ajamalla se kymmenellä eri syöttöaineistolla VERKKO.TXT ja vertaamalla tuloksia oikeisiin.

Jokaisella syöttöaineistolla toimitaan seuraavasti. Aluksi ohjelmasi SOKKELO.EXE kopioidaan hakemistoon, jossa on asianmukainen syöttötiedosto. Tämän jälkeen ohjelmasi suoritetaan tässä hakemistossa.

Jos suoritus katkeaa ajonaikaiseen virheeseen, et saa tästä syöttötiedostosta pisteitä. Muussa tapauksessa ohjelmasi tulostiedostot POLUT.TXT ja KAARI.TXT arvostellaan erikseen. Jos suoritus päättyy määräajassa, kumpikin tiedosto arvostellaan.

Määräaika on 20 sekuntia.

Jos suoritus ei pääty määräajassa, sen suoritus katkaistaan määräajan loputtua, ja tiedostoa KAARI.TXT ei arvostella, vaikka ohjelmasi sellaisen olisi tulostanutkin. Tiedosto POLUT.TXT arvostellaan, mikäli ohjelmasi on ehtinyt tulostaa sen ennen suorituksen katkeamista.

Tiedostosta POLUT.TXT löytyvä ensimmäinen polku antaa kolme pistettä, jos se on lyhin mahdollinen. Toinen polku antaa samoin kolme pistettä, jos se on arvoltaan suurin. Polut arvostellaan toisistaan riippumatta, joten on mahdollista saada toisesta pisteet, vaikka toinen olisikin väärin.

Tiedosto KAARI.TXT antaa neljä pistettä, jos se on kokonaisuudessaan oikein. Tästä ei voi saada osapisteitä.

Yhteensä voit siis saada kymmenen pistettä kustakin syöttöaineistosta, eli kaikkiaan 100 pistettä.