

Datatahti 2000: alkukilpailun ohjelmointitehtävä

1 Lyhyt tehtävän kuvaus

Tehtävänä on etsiä puurakenteen esiintymiä kirjaintaulukosta. Ohjelmasi saa syötteenä kirjaintaulukon ja puun, jonka jokaisessa solmussa on jokin kirjain; katso esimerkkiä kuvassa 1. Tulosteena on annetun puun *esiintymä* kirjaintaulukossa; katso esimerkkiä kuvassa 2. Puun esiintymä siis liittyy jokaiseen puun solmuun yhden kirjaintaulukon alkion siten, että

- solmun ja taulukon kirjaimet ovat samat ja
- solmuihin liitettyjen taulukon alkioiden sijainnit toisiinsa nähden ovat pääpiirteissään samat kuin itse solmujen sijainnit puussa.

Tarkemmin sanoen tehtävänä on etsiä annetulle puulle sellainen esiintymä, että puu on vaakasuunnassa mahdollisimman kapea. Tarvittavien peruskäsitteiden määritelmät kerrataan luvuissa 2 ja 3.

2 Puiden peruskäsitteet

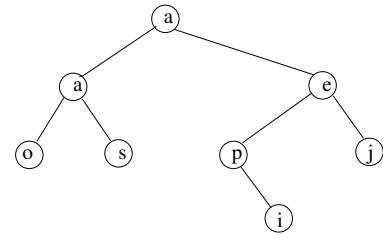
Tehtävässä käsitellään puumaisia rakenteita, joita tietojenkäsittelyssä kutsutaan binääripuiksi. Kuvassa 3 on esimerkki binääripuusta. Puurakenne koostuu *solmuista* ja niiden välisistä yhteyksistä. Jatkossa ajatellaan puun solmut numeroituiksi $1, \dots, k$, missä k on puun koko.

Puurakenne muodostuu solmuista ja niiden välisistä yhteyksistä seuraavasti:

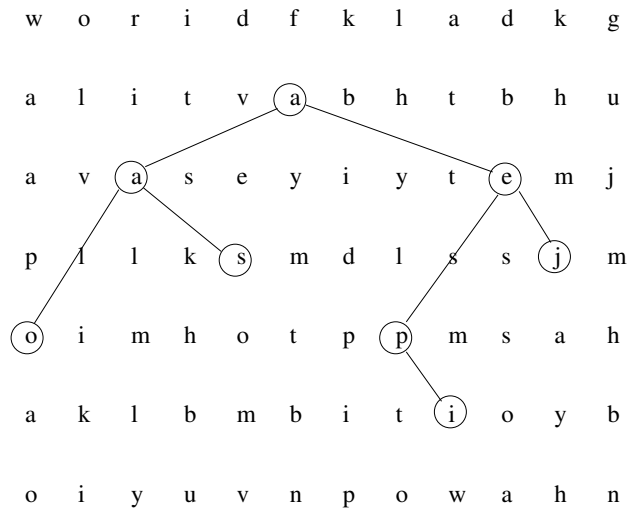
- Yksi solmuista on erityisasemassa puun *juurena*; juuri piirretään kuvassa ylimmäksi solmuksi.
- Juurta lukuunottamatta jokaisella solmulla on tasan yksi *vanhempi*; kuvassa solmu piirretään vanhempaansa alemmaksi ja yhdistetään vanhempaansa viivalla.

Siis kuvan 3 puussa on yhdeksän solmua, joista juurena on solmu 3. Solmu 4 on solmujen 8 ja 1 vanhempi, ja solmu 9 on solmun 7 vanhempi. Jos solmu i on solmun j vanhempi, sanomme että solmu j on solmun i *lapsi*.

w o r i d f k l a d k g
 a l i t v a b h t b h u
 a v a s e y i y t e m j
 p l l k s m d l s s j m
 o i m h o t p p m s a h
 a k l b m b i t i o y b
 o i y u v n p o w a h n



Kuva 1: Esimerkki kirjaintaulukosta ja puusta



Kuva 2: Esimerkki puun esiintymästä kirjaintaulukossa

Binääripuu on puurakenne, jossa kullakin solmulla voi olla korkeintaan kaksi lasta, ja nämä lapset on nimetty erikseen *vasemmaksi* ja *oikeaksi lapseksi*. Binääripuun solmulla voi siis olla

- sekä vasen että oikea lapsi (kuvan 3 solmut 2, 4, ja 6),
- vasen lapsi mutta ei oikeaa (kuvan 3 solmu 9),
- oikea lapsi mutta ei vasenta (kuvan 3 solmu 3) tai
- ei lainkaan lapsia (kuvan 3 solmut 1, 5, 7 ja 8).

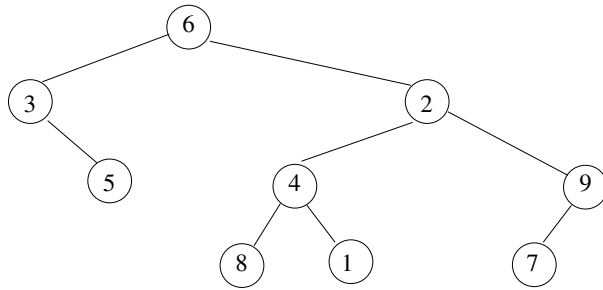
Binääripuun solmun *vasemman alipuuhun* kuuluu tämän solmun vasen lapsi sekä vasemman lapsen kaikki lapset, lapsen lapset jne. Jos solmulla ei ole vasenta lasta, sanotaan että sen vasen alipuu on *tyhjä*. Vastaavasti määritellään *oikea alipuu*. Siis kuvan 3 binääripuun solmun 6 vasemmassa alipuussa on solmut 3 ja 5 ja oikeassa alipuussa solmut 1, 2, 4, 7, 8 ja 9. Solmun 2 vasemmassa alipuussa on solmut 1, 4 ja 8 ja oikeassa alipuussa solmut 7 ja 9. Solmun 3 oikeassa alipuussa on solmu 5; vasen alipuu on tyhjä.

3 Binääripuun esiintyminen kirjaintaulukossa

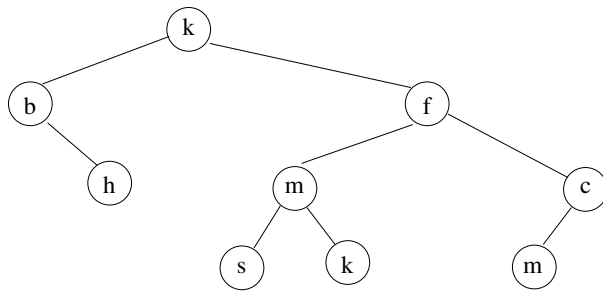
Tarkastellaan nyt binääripuuta, jonka kuhunkin solmuun on lisäksi liitetty jokin kirjain. Olkoon k binääripuun solmujen lukumäärä, ja oletetaan, että taulukko $C[1..k]$ antaa kuhunkin solmuun liittyvän kirjaimen. Merkitään solmuun i liittyvää kirjainta $C[i]$. Kuvassa 4 on kuvan 3 binääripuuhun merkitty tällaiset kirjaimet.

Oletetaan lisäksi annetuksi kaksiulotteinen kirjaintaulukko $M[1..n, 1..m]$. Taulukko ajatellaan asetelluksi niin, että kirjain $M[1, 1]$ on vasemmassa yläkulmassa, merkki $M[n, 1]$ vasemmassa alakulmassa ja merkki $M[n, m]$ oikeassa alakulmassa. Binääripuun *sijoittelu* tällaiseen taulukkoon liittyy jokaiseen puun solmuun yhden taulukon alkion. Toisin sanoen sijoittelu on taulukko $L[1..k]$, jonka alkiot ovat taulukon M indeksejä eli pareja $(i, j) \in \{1, \dots, n\} \times \{1, \dots, m\}$. Sijoittelu on binääripuun *esiintymä*, jos seuraavat ehdot ovat voimassa:

1. Solmussa on aina sama kirjain kuin siihen liitettyssä taulukon alkiossa; toisin sanoen $M[i, j] = C[s]$ kun $L[s] = (i, j)$.
2. Solmun vasen alipuu on kokonaisuudessaan solmusta alavasemmalle; toisin sanoen jos solmu t on solmun s vasemmassa alipuussa ja $L[s] = (i_1, j_1)$ ja $L[t] = (i_2, j_2)$, niin $i_1 < i_2$ ja $j_1 > j_2$.
3. Solmun oikea alipuu on kokonaisuudessaan solmusta alaoikealle; toisin sanoen jos solmu t on solmun s oikeassa alipuussa ja $L[s] = (i_1, j_1)$ ja $L[t] = (i_2, j_2)$, niin $i_1 < i_2$ ja $j_1 < j_2$.



Kuva 3: Esimerkki binääripuusta



Kuva 4: Kirjaimilla merkitty binääripuu

Binääripuun sijoittelun *leveys* on sen vaatiman vasemmanpuoleisimman sarakkeen ja oikeanpuoleisimman sarakkeen väliin jäävien sarakkeiden lukumäärä (nämä sarakkeet mukaanlukien). Kuvan 2 sijoittelun leveys on siis 11, koska vasemmanpuoleisin solmu (solmu 3 jossa 'o') on sarakkeessa 1 ja oikeanpuoleisin solmu (solmu 8 jossa 'j') on sarakkeessa 11. Edellä esitetyn matemaattisen määritelmän mukaan siis binääripuun sijoittelun L leveys on siis

$$\max S - \min S + 1$$

missä

$$S = \{j \mid (i, j) = L[k] \text{ joillakin } i \text{ ja } k\} .$$

4 Syötteen

Ohjelmasi tulee lukea syöteensä oletushakemiston tekstitiedostosta `input.txt`. Tiedoston sisältö on seuraava:

- Ensimmäisellä rivillä on puun solmujen lukumäärä k . Voit olettaa, että $1 \leq k \leq 50$.
- Seuraavilla k rivillä on kullakin puun yhden solmun kuvaus muodossa $c t u$, missä c on englannin kielen aakkoston pieni kirjain (jokin merkeistä `a, b, c, ..., z`) ja t ja u ovat kokonaislukuja, $1 \leq t \leq k$ ja $1 \leq u \leq k$. Merkki c ja kokonaisluku t , samoin kuin kokonaisluku t ja kokonaisluku u , on erotettu toisistaan yhdellä välilyönnillä. Rivillä $s + 1$ oleva syöte $c t u$ tarkoittaa, että solmuun s liittyvä kirjain on c , solmun s vasen lapsi on solmu t ja oikea lapsi u . Jos solmulla ei ole vasenta lasta, niin $t = 0$; jos solmulla ei ole oikeaa lasta, niin $u = 0$.
- Rivillä $k + 2$ on yhdellä välilyönnillä erotetut kaksi kokonaislukua n ja m , jotka ovat kirjaintaulukon rivien ja sarakkeiden lukumäärä. Voit olettaa, että $1 \leq 20 \leq n$ ja $1 \leq 200 \leq m$.
- Seuraavat n riviä sisältävät kirjaintaulukon kirjaimet ilman välilyöntejä. Rivin $k + i + 2$ merkki j on taulukon alkio $M[i, j]$. Nämä alkiot ovat englannin kielen aakkoston pieniä kirjaimia.

5 Tulosteet

Ohjelmasi tulosteena tulee olla oletushakemistossa tekstitiedosto `output.txt`. Tiedoston loppua pitää edeltää rivinvaihto, mutta muuten tiedostossa ei saa olla muita rivinvaihtoja ja välilyöntejä kuin mitä alla on sanottu.

Jos annetulla binääripuulla ei ole esiintymää annetussa kirjaintaulukossa, tulostiedostossa pitää olla vain yksi rivi ja sillä vain yksi numero 0.

Jos binääripuulla on esiintymä kirjaintaulukossa, tulostiedostossa tulee olla tasan k riviä ja jokaisella rivillä kaksi kokonaislukua yhdellä välilyönnillä erotettuna. Rivillä s olevat kokonaisluvut antavat leveydeltään mahdollisimman pienessä esiintymässä solmuun s liitetyn taulukon alkion rivin ja sarakkeen. Toisin sanoen jos L on tämä leveydeltään pienin esiintymä, tulostiedoston rivillä s tulee olla i j missä $L[s] = (i, j)$.

Jos annetulla binääripuulla on useita leveydeltään minimaalisia esiintymiä annetussa kirjaintaulukossa, sinun pitää esittää tulostiedostossa mikä tahansa yksi niistä.

6 Esimerkki

Seuraavassa on eräs kuvan 1 esimerkkiä vastaava tiedosto `input.txt`.

```
8
a 2 5
a 3 4
o 0 0
s 0 0
e 6 8
p 0 7
i 0 0
j 0 0
7 12
woridfkladkg
alitvabhtbhu
avaseiytemj
pllksmdlssjm
oimhotppmsah
aklbmbitiyob
oiyuvnpowahn
```

Kuvan 2 mukaista esiintymää vastaava tiedosto `output.txt` olisi seuraava:

```
2 6
3 3
5 1
4 5
3 10
5 8
6 9
4 11
```

7 Ratkaisun palautus ja arvostelu

Tehtävän ratkaisuna sinun pitää palauttaa asianmukaisesti kommentoitu **lähdekielinen** ohjelma. Sallittuja ohjelmointikieliä ovat Pascal, C ja C++. Laita ohjelmasi alkuun kommentiksi yhteystietosi ja mahdolliset ohjeet ohjelman kääntämistä varten. Ohjelmalla pitää olla vain edellä määritellyt syötteet ja tulosteet. Erityisesti sen ei pidä lukea mitään näppäimistöä tai tulostaa mitään näyttöruudulle.

Arvosteluvaiheessa ohjelmasi käännetään (Unix-ympäristössä) ja suoritetaan useilla erilaisilla testisyötteillä. Pisteitä saa käännetyn ohjelman oikeasta ja nopeasta toiminnasta joukolla testisyötteitä. Lähdekielistä ohjelmaa ei siis arvostella, kunhan vain sen käänнос onnistuu. Siirrettävyysohjelmien välttämiseksi pitäydy kielen peruspiirteissä.

Testausympäristössä on kääntäjinä käytettävissä ainakin Gnu Pascal ja C/C++, Free Pascal (jonka pitäisi olla Turbo Pascal -yhteensopiva; ks. <http://deadlock.et.tudelft.nl/~fpc/>) sekä Sun Workshop C, C++ ja Pascal (jokseenkin standardien mukaisia kaupallisia Unix-toteutuksia). Käännökset tehdään kääntäjien oletusarvoisilla optioilla. Jos käytettävissä olevista kielen piirteistä tms. on epäselvyyttä, kysy asiasta sähköpostitse (Jyrki.Kivinen@cs.Helsinki.FI). Seuraa myös sivua <http://www.cs.Helsinki.FI/~jkivinen/datatahti/ohjeet.html>, jonne tulee tarpeen mukaan lisäohjeita.

Testisyötteistä osa testaa vain ohjelman oikeellisuutta. Kustakin tällaisesta syötteestä saa tietyn määrän pisteitä, jos ohjelma tuottaa oikean tuloksen kohdullisessa ajassa. Syötteiden koot ja aikarajat valitaan siten, että hitaallakin ohjelmalla saa pisteitä, kunhan se on muuten toimiva. Osa testisyötteistä testaa ohjelman tehokkuutta. Näillä syötteillä ohjelman saama pistemäärä vähenee sitä mukaa, kun suoritus aika kasvaa. Testisyötteet ovat aina tarkalleen ylläolevan kuvauksen mukaisia; virheellisillä ohjelmalla saa tehdä mitä tahansa (vaikka pysähtyä ajonaikaiseen virheeseen). Pisteytyksen yksityiskohtia ja tarkkoja aikarajoja ei tässä vaiheessa julkisteta. Vertailukohdaksi voidaan kuitenkin antaa, että tehtävän laatijan ratkaisu vaatii alle 500 kilotavua muistia ja vie suurimmilla testisyötteillä noin yhden sekunnin suoritus aikaa (Free Pascal, Linux, 166 MHz Pentium).

Palauta ratkaisusi mieluiten sähköpostiviestinä (ei liitetiedostona) osoitteeseen Jyrki.Kivinen@cs.Helsinki.FI. Älä käytä yli 80 merkin rivejä! Voit myös toimittaa ratkaisusi 3,5 tuuman DOS-formatoidulla levykkeellä osoitteeseen

Jyrki Kivinen
Tietojenkäsittelytieteen laitos
PL 26 (Teollisuuskatu 23)
00014 HELSINGIN YLIOPISTO.

Pakkaa levyke huolellisesti ja laita mielellään toinen samansisältöinen mukaan varmuuskopioksi. Talleta ohjelmasi juurihakemistoon nimellä `dt2000.pas`, `dt2000.c` tai `dt2000.cpp`.