

Programozási nyelvek 2

Kiss Endre Farkas

(KIELAAT.elte)

2006/07 / Tavaszi szemeszter

Gyakorlatvezető: Papp-Varga Zsuzsanna

C++ beadandó

Tartalomjegyzék:

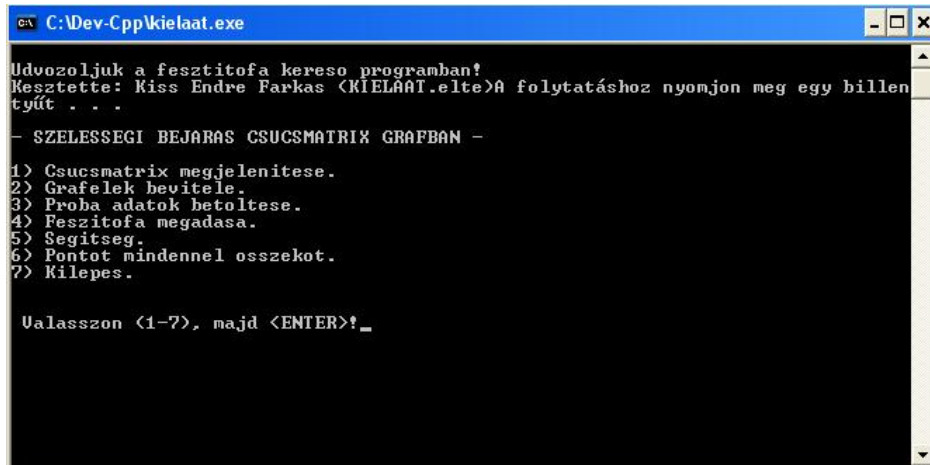
C++ beadandó	1
Tartalomjegyzék:	1
A program feladata:	2
Felhasználói leírás:	2
Gráfélek bevitele:	2
Csúcsmátrix megjelenítése:	2
Próba adatok betöltése	3
Segítség	3
A program algoritmus	3
Sorműveletek:	3
Halmazműveletek:	3
Gráfműveletek:	3
Próba futtatás	5

A program feladata:

C++ program, az alábbi feladatra: Adott N város közötti úthálózat csúcsmátrixszal ábrázolva. Határozzuk meg a gráf egy feszítőfáját! Olvassuk be a csúcsmátrixot, majd jelenítsük meg az eredményt.

Felhasználói leírás:

Induláskor a program alapvető funkcióiról tájékoztató menüt láthatunk.



```
C:\Dev-Cpp\kielaaat.exe
Udvozoljuk a feszitofa kereso programban!
Kesztette: Kiss Endre Farkas (KIELAAI.elte)A folytatáshoz nyomjon meg egy billen
tyút . . .
- SZELESSEGI BEJARAS CSUCSMATRIX GRAFBAN -
1> Csucsmatrix megjelenitese.
2> Grafek bevitete.
3> Proba adatok betoltese.
4> Feszitofa megadasa.
5> Segitseg.
6> Pontot mindennel osszekot.
7> Kilepes.

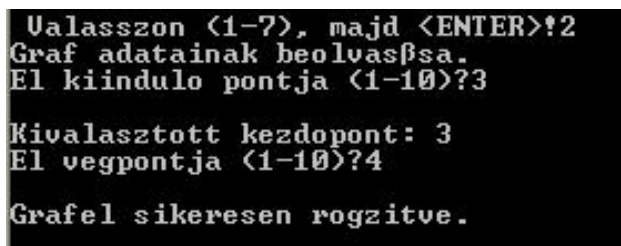
Ualasszon <1-7>, majd <ENTER>!_
```

Gráfélek bevitele:

Új gráfél hozzáadása a csúcsmátrixhoz.

A rendszer először megkérdezi, hogy mi az kiinduló pontja, majd pedig bekéri, hogy mi legyen annak a végpontja.

Mivel az egyes városok között oda-vissza is lehet közlekedni, a visszaút is megjelenik a csúcsmátrixban.



```
Ualasszon <1-7>, majd <ENTER>!2
Graf adatainak beolvasása.
El kiindulo pontja <1-10>?3
Kivalasztott kezdopont: 3
El vegpontja <1-10>?4
Grafel sikeresen rogzitve.
```

Csúcsmátrix megjelenítése:

Megjeleníti szemléletesen a csúcsmátrix adatait a képernyőn, majd újra megjeleníti a menüpontokat.

```
Valasszon <1-7>, majd <ENTER>#1
Csucsmatrix
(- - nincs kapcsolat, T - van kapcsolat)
- T T - - - - -
T - T T - T - - - -
T T - T T - - - - -
- T T - T T - - - - -
- - T T - T - - - - -
- T - T T - - - - -
- - - - - - - - - -
- - - - - - - - - -
- - - - - - - - - -
- - - - - - - - - -
```

Proba adatok betöltése

A csúcsmátrixot feltölti próba adatokkal, majd erről tájékoztatást ad. Végül újra kiírja a választható menüpontokat.

```
Valasszon <1-7>, majd <ENTER>#3
Proba adatok sikeresen betolteve.
```

Segítség

Rövid szöveges tájékoztatást jelent meg a program funkcióiról.

```
- SEGITSEG -
Ennek a programnak az a feladata, hogy egy
csucsmatrix-szal abzazolt grafnak adja meg
egy feszitofajat szelessegi bejaras alapjan.
A program kezelesi a kezdeti menupontok alapjan.
```

A program algoritmus

A program logikai (IGEN, NEM elemeket tartalmazó felsorolás) elemekből álló vektor segítségével ábrázolja a halmaz, a sor, és a gráf-műveleteket.

Sorműveletek:

- **SorUresE()** – logikai értéként visszaadja, hogy a sor üres-e
- **SorTeleE()** – logikai értéként visszaadja, hogy a sor tele van-e
- **Sorba(PontTip)** – adott pontot berakja a sorba
- **Sorbol()** – egy pontot ad vissza, amit a sorból szed ki
- **SorHibasE()** – logikai értéként visszaadja, hogy van-e valami hiba a sorban

Halmazműveletek:

- **HalmazUres()** – kiüríti a halmazt
- **Halmazba(PontTip)** – a paraméterül kapott pontot a halmazba rakja
- **Eleme(PontTip)** – logikai értéként visszaadja, hogy egy adott pont eleme-e a halmaznak

Gráf-műveletek:

- **GrafUres()** kiüríti a gráfot

- **Osszekot** (PontTip, PontTip) – összeköti a paraméterül kapott pontokat
- **Elszakit** (PontTip, PontTip) – az adott pontok közötti élt szakítja szét
- **vanEl** (PontTip, PontTip) – logikai értéként visszaadja, hogy van-e két két pont között
- **SzomszedPontokSzama**(PontTip) – adott pont szomszédjainak a számát adja vissza
- **minimumPont()** – a legkisebb indexű elemet adja vissza, amiből indul el
- **SzomszedPont**(PontTip, int) – Adott pont adott szomszédját adja vissza

A szélességi bejárás algoritmus szintén a klasszikus bejárési algoritmus változatlan adoptálása.

A nekünk leginkább megfelelő feszítőfa megtalálása az érdekes, mely az alábbi algoritmus szerint történik:

```

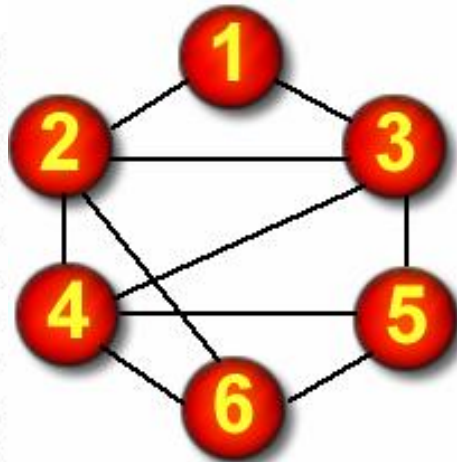
kezdes=helyigraf.minimumPont(); //A legkisebb, éllel rendelkező pont
Ha kezdes=11 akkor
    kiír: „HIBA! Nincsenek pontok a gráfban.”
különben
    /* Hány olyan pont van összesen a gráfban,
    melyből indul ki él */
    pontokSzama= pontSzam();
    // Minden kezdponttal megpróbálunk generálni feszítőfát
    maxFeszitofa=0;
    maxFeszitofaIndex=1;
    // Megkeressük azt a pontot, ahonnan indulva a bejárás a legtöbb pontot érinti
    Ciklus i=1-től 10-ig
        eredmény=szelessegiBejaras(i);
        Ha eredmény.pontSzam>maxFeszitofa akkor
            maxFeszitofa=eredmény.pontSzam;
            maxFeszitofaIndex=i;
        Elágazás vége
    Ciklus vége
    // Végül a legtöbb pontot érintő eredményt vesszük alapul
    eredmény=szelessegiBejaras(maxFeszitofaIndex);
    Ha eredmény.pontSzam=pontokSzama akkor
        eredményKiir(eredmény);
        Kiír: „Sikeres bejárás”
    különben
        kiír: „Sajnos vannak a gráfban izolált pontok
    Elágazás vége
Elágazás vége

```

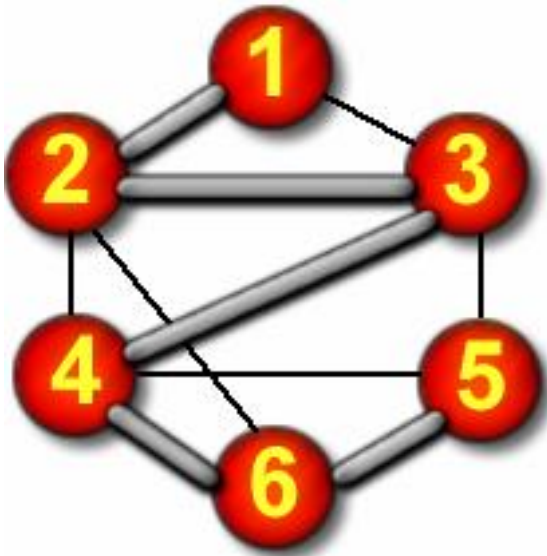
Próba futtatás

A programra épített próbaadatok az alábbi gráfot ábrázolják:

```
Osszekot (1,2) ;  
Osszekot (1,3) ;  
Osszekot (2,3) ;  
Osszekot (2,4) ;  
Osszekot (3,4) ;  
Osszekot (2,6) ;  
Osszekot (3,5) ;  
Osszekot (4,5) ;  
Osszekot (4,6) ;  
Osszekot (5,6) ;
```



Erre a program az alábbi feszítőfát találja meg:



```
Feszitofa szelessegi bejarassal:  
A megtalalt feszitofa:  
1 2 3 4 6 5  
Sikeres bejaras.
```

Ez az eredeti gráf részgráfja, tartalmazza minden pontját, tehát definíció szerint feszítőfa. Megállapítható tehát, hogy a próbaadatokra helyes kimenetet produkált a program.