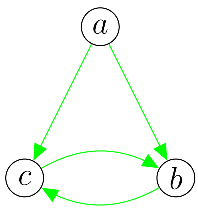
Algoritmusok és adatszerkezetek gyakorlat – 10

Gráfalgoritmusok I.

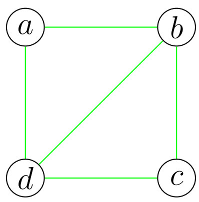
Gráf

Egy $G = (V, E)$ struktúrát gráfnak nevezünk, ahol:

- V a csúcsok halmaza.
- $E \subseteq V \times V$ az élek halmaza, vagyis csúcspárok
  - ha ez egy rendezett pár, akkor azt mondjuk, hogy a gráf irányított
  - ha nem rendezett (tehát ha (u, v) létezik, akkor (v, u) is), akkor pedig irányítatlan gráfról beszélünk
- Egy irányítatlan gráf összefüggő, ha bármely két csúcs között van út (u-ból v-be van út, ha nulla, egy vagy több egymás utáni él követésével u-ból eljuthatunk v-be).
- Egy irányított gráf erősen összefüggő, ha bármely két csúcs között van irányított út.
- Egy $G = (V, E, C)$ gráf súlyozott, ha minden élhez egy címkét/súlyt rendelünk $C : E \to$ súly, azaz C az élekhez súlyt rendel (a megengedett súlyok vagy címkék halmazából)
- Egy irányított $G = (V, E)$ gráf transzponáltja: $G^T = (V, E^T)$, ahol $E^T = \{(p, q) : (q, p) \in E\}$. (Tehát megfordítjuk az élek irányítását.)



Irányított gráf

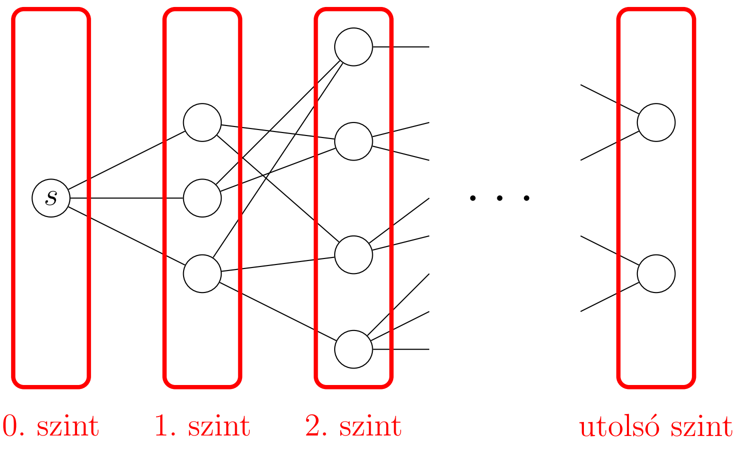


Irányítatlan gráf

Szélességi keresés

A gráf szintről szintre való feltérképezése egy s kezdőcsúcsból.

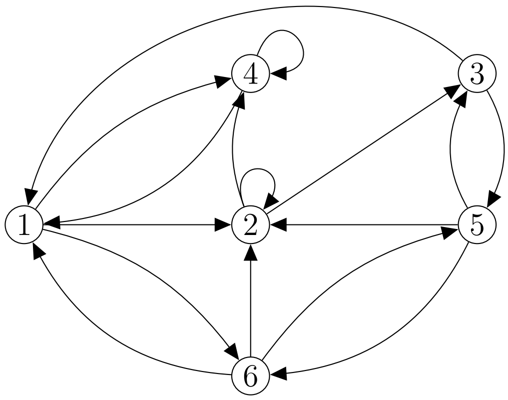
- 0. szint: $\{s\}$
- i. szint: azok a csúcsok, amik s-ből i lépésből elérhetőek (de kevesebből nem)



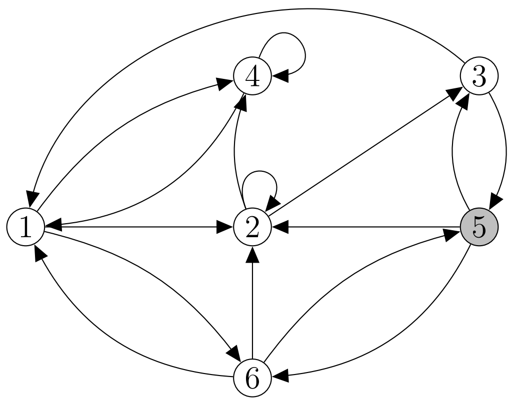
Algoritmus:

```
SZELTKERES(G,s)
  for(p in V){
    Szin(p):=feher
    Apa(p):=-1
    d(p):=INF}
  Szin(s):=szurke
  d(s):=0
  Apa(s):=0
  Letesit(S:Sor)
  Sorba(S,s)
  while(Elemszam(S)>0){
    Sorbol(S,u)
    for(v in Kiel(G,u)){
      if(Szin(v)==feher) {
        Szin(v):=szurke
        Apa(v):=u
        d(v):=d(u)+1
        Sorba(S,v)}}
    Szin(u):=fekete
  }
```

1. Feladat A következő gráfon az 5 csúcsból hajtsuk végre a szélességi keresést:



Megoldás



csúcs	1	2	3	4	5	6
Apa	-1	-1	-1	-1	0	-1
d	∞	∞	∞	∞	0	∞

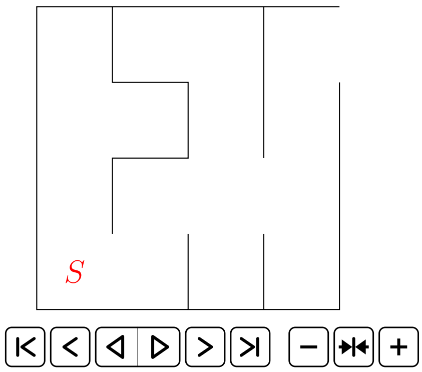
S	5					

Tehát a 0. szinten az 5, az első szinten a 2, 3, 6, a második szinten pedig a 4 és az 1 csúcsok lesznek.

Mélységi keresés

Olyan, mint kijutni egy labirintusból:

- követed az utat, amíg nem jutsz el egy akadályig
- backtrack visszafelé a kenyérmorzsák mentén, amíg egy eddig meg nem látogatott szomszédot nem találsz
- rekurzívan feltárod ezt a szomszédot is
- ügyelsz közben arra, hogy már meglátogatott pontba ne menj vissza még egyszer



Algoritmus:

```
Melykeres(G)
  for(u in G){
    szin(u):=feher
    Apa(u):=0
  }
  ido:=0
  for(u in G){
    if(szin(u)=feher)
      MBejar(u)
  }
```

```
MBejar(u)
  szin(u):=szurke
  ido:=ido+1
  d(u):=ido
  for(v in Kiel(G,u)){
    if(szin(v)=feher){
      Apa(v):=u
      MBejar(v)}}
  szin(u):=fekete
  ido:=ido+1
  f(u):=ido
```

Az elérési ($d(u)$) és elhagyási ($f(u)$) idők számítása nélkül is helyes az algoritmus. Utóbbira a mélységi keresést felhasználó algoritmusok esetében lehet szükség.

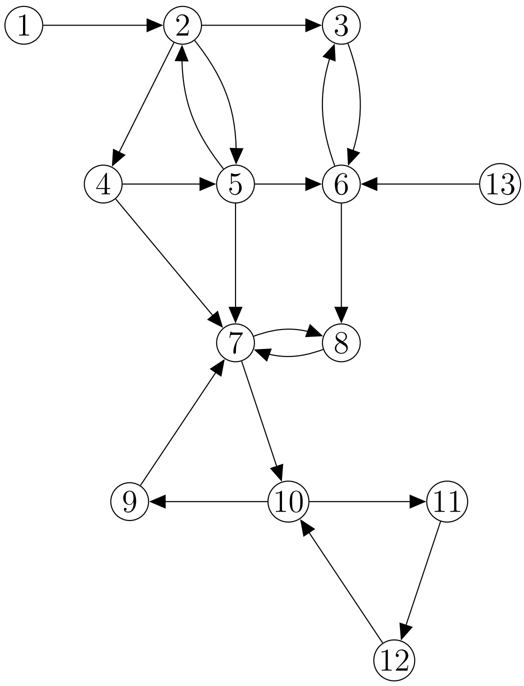
Az algoritmus egy ún. mélységi feszítőerdőt (MFE) ad eredményül.

Élek osztályozása

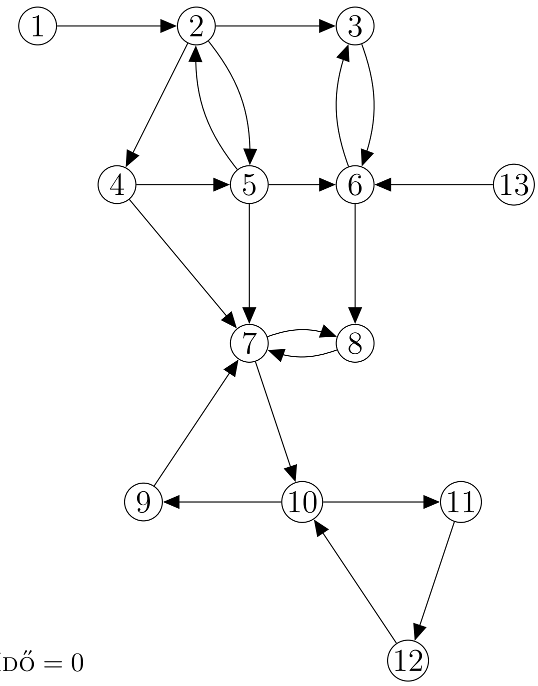
- **Faél:** $(u, v) \in E$ faél, ha bekerül a MFE élei közé, azaz $Apa(v) = u$.
- **Visszaél:** $(u, v) \in E$ visszaél, ha u leszármazottja v-nek a MFE-ben.
- **Előreél:** $(u, v) \in E$ előreél, ha v leszármazottja u-nak a MFE-ben és nem faél.
- **Keresztél:** Minden más esetben $(u, v) \in E$ keresztél.

Lemma: Egy irányított gráfban akkor és csak akkor van kör, ha a mélységi feszítőerdejében van visszaél.

2. Feladat A következő gráfon hajtsuk végre a mélységi keresést! Osztályozzuk az éleket!



Megoldás



IDŐ = 0

Csúcs	1	2	3	4	5	6	7	8	9	10	11	12	13
Apa	0	0	0	0	0	0	0	0	0	0	0	0	0
d	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
Szín	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯
f	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞

Erősen összefüggő komponensek

Erősen összefüggő komponensek: a gráfban azok a maximális csúcshalmazok, amin belül bármelyik csúcsból el lehet jutni bármely másikba.

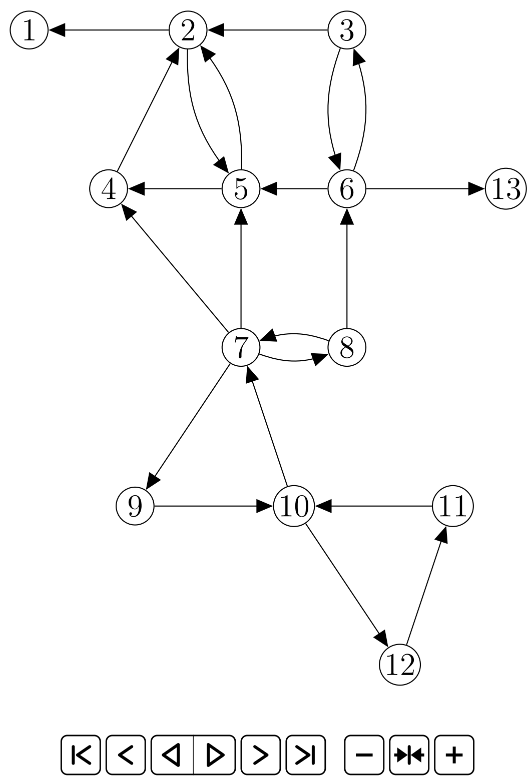
Meghatározása:

1. Számítsuk ki a MELYKERES algoritmussal az $f(u)$ elhagyási értékeket.
2. A G^T transzponált gráfra alkalmazzuk a MELYKERES eljárást úgy, hogy a pontokra a MBEJAR eljárást f szerint csökkenő sorrendben hívjuk.
3. A 2. pontban az egy mélységi feszítőfába kerülő pontok alkotnak egy erősen összefüggő komponenst.

3. Feladat Határozzuk meg az előző feladatban adott gráf erősen összefüggő komponenseit!

Megoldás

Transzponáljuk a gráfot, azaz fordítsuk meg az élei irányítását, majd hajtsuk végre a Mélységi bejárást a csúcsokból a $13, 1, 2, 5, 4, 3, 6, 8, 7, 10, 11, 12, 9$ csúcsokból.



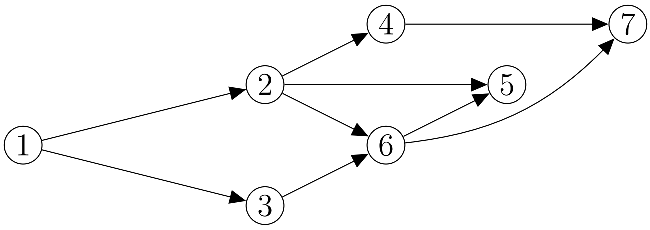
Topologikus rendezés

Egy $G = (V, E)$ irányított körmentes gráf (Directed acyclic graph, DAG) topologikus rendezésén a V pontjainak egy olyan $v_1, v_2, \ldots, v_n$ ($n = |V|$) felsorolását értjük, amelyre teljesül, hogy minden $(u, v) \in E$ élre, u előbb van a felsorolásban, mint v.

Meghatározása:

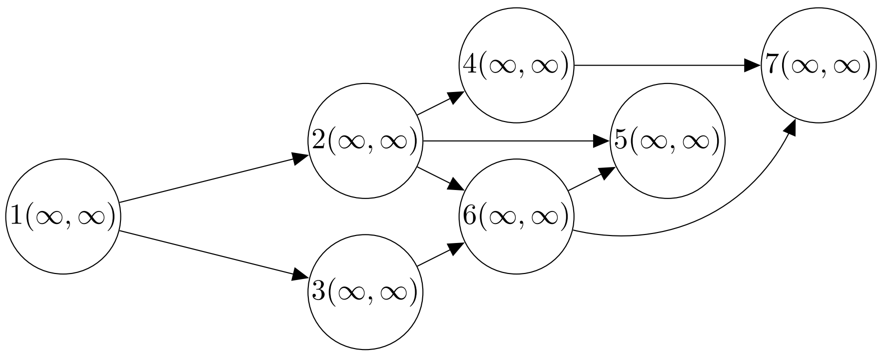
1. A mélységi keresés algoritmusát hajtsuk végre a gráfra.
2. Az egyes csúcsok elhagyásakor beszúrjuk őket egy láncolt lista elejére.
3. A csúcsok láncolt listája adja meg a rendezési sorrendet.

4. Feladat Határozzuk meg a következő gráf topologikus rendezését!



Megoldás

Az egyszerűség kedvéért most a csúcsok mellé jegyezzük fel a d és f értékeket. Az elért csúcsok színe szürke, az elhagyottaké most piros lesz.



Lista:
