Gráfok
1. előadás

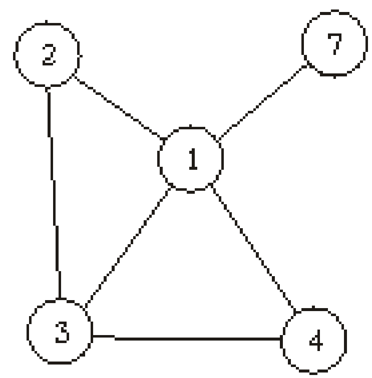
Gráfok

A gráf fogalma:

Gráf(P,E): P *pontok* (*csúcsok*) és $E \subseteq P \times P$ *élek* halmaza

Fogalmak:

- Irányított gráf : $(p_1,p_2) \in E$-ből nem következik, hogy $(p_2,p_1) \in E$
- Irányítatlan gráf : $(p_1,p_2) \in E \rightarrow (p_2,p_1) \in E$
- Út: $(p_1,p_2), (p_2,p_3), \ldots, (p_{k-1},p_k) \in E$ élsorozat
- Kör : $(p_1,p_2), (p_2,p_3), \ldots, (p_{k-1},p_1) \in E$ élsorozat
- Körmentes gráf: E-ben nincs kör
- Hurokél: $(p,p) \in E$



Gráfok

Fogalmak:

- Fok: p∈P-hez csatlakozó élek száma irányítatlan gráfban
- Befok, Kifok: egy p∈P pontba bevezető, illetve kivezető élek száma irányított gráfban
- Összefüggő gráf: ∀p,q∈P: ∃út(p,q) – irányítatlan gráf
- Erősen összefüggő gráf: ∀p,q∈P: ∃út(p,q) – irányított gráf
- Összefüggő komponens: (R,F)⊆(P,E) összefüggő gráf
- Erősen összefüggő komponens: (R,F)⊆(P,E) erősen összefüggő irányított gráf

- Összefüggő gráf: ∃p∈P: ∀q∈P: ∃út(p,q) – irányított gráf

Gráfok

Fogalmak:

- Súlyozott gráf: ($P,E,s:E\rightarrow R$ mérték), (P-hez is lehet súly)
- Fa: összefüggő körmentes gráf
- Erdő (liget): körmentes gráf
- Feszítőfa: a gráf összes pontját tartalmazó fa
- Forrás: irányított gráf pontja, amelyből csak kivezető él van
- Nyelő: irányított gráf pontja, amelybe csak bevezető él van
- Háló: körmentes irányított gráf, egy forrással és nyelővel
- Izolált pont: legfeljebb hurokél kapcsolódik hozzá

Gráfok ábrázolása

Csúcsmátrix (szomszédsági mátrix)**:**

$$C(i,j)=\begin{cases} igaz & ha \quad (i,j)\in E \\ hamis & ha \quad (i,j)\notin E \end{cases}$$

Súlyozott gráfra:

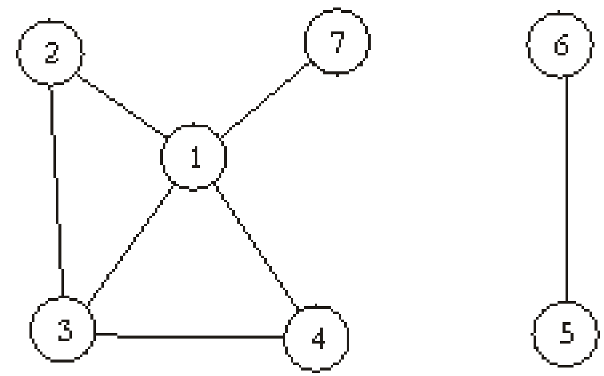
$$C(i,j)=\begin{cases} s(i,j) & ha \quad (i,j)\in E \\ Nemdef & ha \quad (i,j)\notin E \end{cases}$$

Megjegyzés: *Nemdef*= 0 vagy -1 vagy +∞ vagy …

Gráfok ábrázolása

Csúcsmátrix:

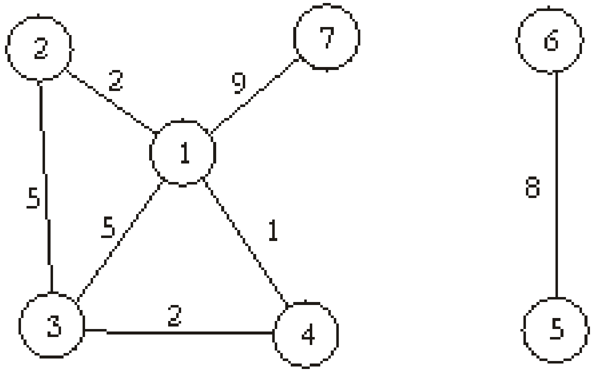
	i	i	i			i
i		i				
i	i		i			
i		i				
					i	
				i		
i						



Gráfok ábrázolása

Csúcsmátrix súlyozott gráfra:

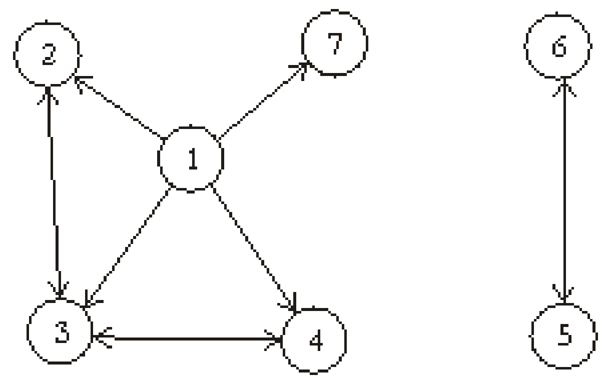
	2	5	1			9
2		5				
5	5		2			
1		2				
					8	
				8		
9						



Gráfok ábrázolása

Csúcsmátrix irányított gráfra:

	i	i	i			i
		i				
	i		i			
		i				
					i	
				i		



Gráfok ábrázolása

Tapasztalatok a csúcsmátrixról:

- irányítatlan gráf esetén szimmetrikus
- irányított gráf esetén nem feltétlenül szimmetrikus
- Fok, Kifok: soronként az igaz értékek száma
- Befok: oszloponként az igaz értékek száma
- könnyű új éleket hozzávenni, éleket törölni, élek súlyát megváltoztatni
- nehéz új pontokat hozzávenni, pontokat törölni
- kevés él esetén memória-pazarló

Gráfok ábrázolása

Csúcslista (szomszédsági lista) – tömbös megvalósításban:

Fok(i), Kifok(i)= az i-ből kivezető élek száma

Ki(i,j)= az i-ből kivezető j. él végpontja

Súlyozott gráfra:

Fok(i), Kifok(i)= az i-ből kivezető élek száma

Ki(i,j).pont= az i-ből kivezető j. él végpontja

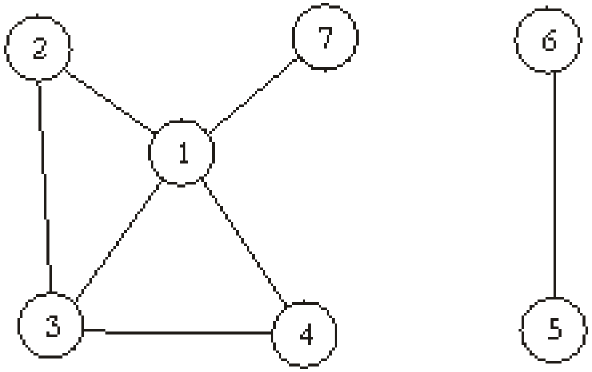
Ki(i,j).súly= az i-ből kivezető j. él súlya

Gráfok ábrázolása

Csúcslista:

4
2
3
2
1
1
1

2	3	4	7
1	3		
1	2	4	
1	3		
6			
5			
1			

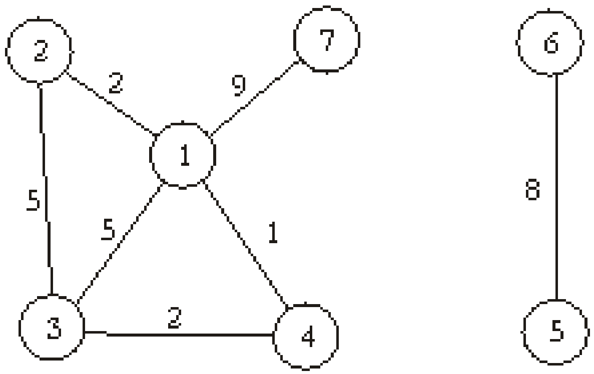


Gráfok ábrázolása

Csúcslista súlyozott gráfra:

4
2
3
2
1
1
1

2,2	3,5	4,1	7,9
1,2	3,5		
1,5	2,5	4,2	
1,1	3,2		
6,8			
5,8			
1,9			

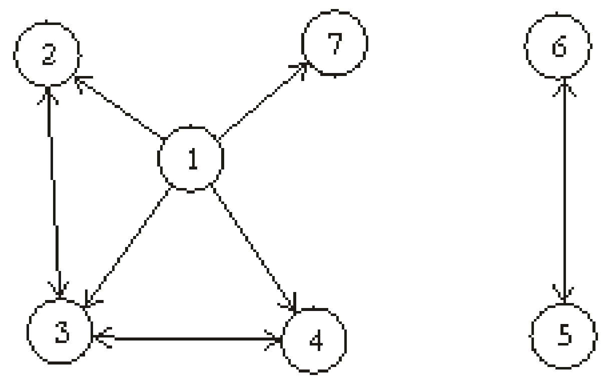


Gráfok ábrázolása

Csúcslista irányított gráfra:

4
1
2
1
1
1
0

2	3	4	7
3			
2	4		
3			
6			
5			



Gráfok ábrázolása

Tapasztalatok a csúcslistáról:

- irányítatlan gráf esetén mindkét végpontnál szerepel a másik
- Fok, Kifok: soronként a darabszám
- Befok: nehezen számítható
- könnyű új éleket hozzávenni, élek súlyát megváltoztatni
- nehéz éleket törölni (sőt irányítatlan gráfnál 2 helyről kell)
- nehéz pontokat törölni
- nagy mátrix kell, ha nincs jó korlát a kivezető élek számára vagy dinamikus méretű tömböket kell használni

Gráfok ábrázolása

Éllista (tömbös megvalósításban):

E(i,j)= az i. él j. végpontja (j=1,2)

Súlyozott gráfra:

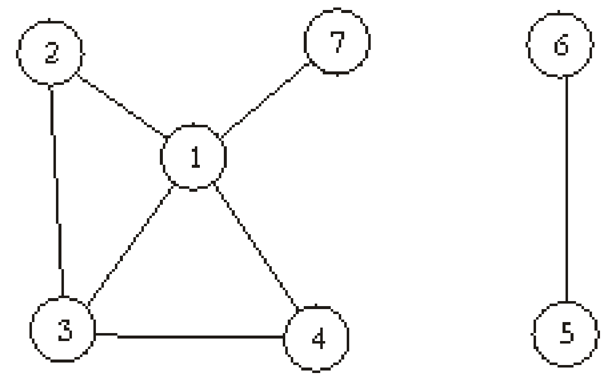
E(i).él(j)= az i. él j. végpontja (*Lehetne: E(i).kezdő, E(i).vég is.*)

E(i).súly= az i. él súlya

Gráfok ábrázolása

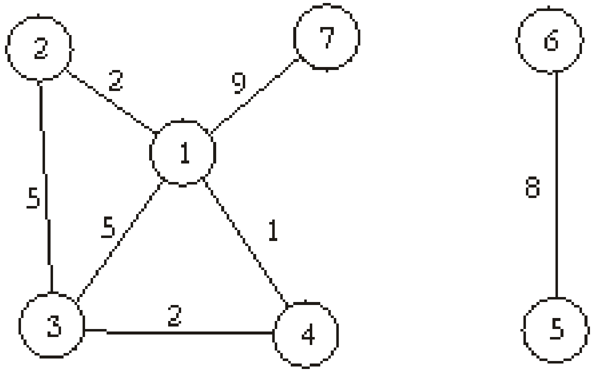
Éllista:

1	2
1	3
1	4
1	7
2	3
3	4
5	6



Éllista súlyozott gráfra:

1	2	2
1	3	5
1	4	1
1	7	9
2	3	5
3	4	2
5	6	8



Gráfok ábrázolása

Tapasztalatok az éllistáról:

- irányítatlan gráf esetén csak egyszer szerepelnek az élek
- könnyű új éleket hozzávenni, élek súlyát megváltoztatni
- nehéz éleket törölni
- nehéz pontokat törölni

Gráfok ábrázolása

A gráftípus (statikus gráf – pontok, élek száma rögzí-tett**):**

Értékhalmaza:

- `Gráf(Sorozat(TÉl:THossz),`
- `          Sorozat(TPont:TElem))`
- `Változó Pontszám, Élszám: Egész`

Műveletei:

- `Érték(Gráf,Pont)`
- `Vanél?(Gráf,Pont1,Pont2)`
- `Élhossz(Gráf,Pont1,Pont2)`

A pontok sorozata sokszor az 1..N számsorozat.

Gráfok ábrázolása

Műveletei:

- Szomszédpontokszáma(Gráf,Pont)
- Szomszéd(Gráf,Pont,i)
- Elsőszomszéd(Gráf,Pont)
- Következőszomszéd(Gráf,Pont)

Speciális műveletek

- Felépít(Gráf1,Gráf2)
  - éllistából csúcsmátrix
  - éllistából csúcslista (tömbös)

A szomszéd súlyozatlan gráf esetén egy pont, súlyozott gráf esetén pedig egy rekord, ami a pontot és az oda vezető él súlyát tartalmazza.

Gráfok ábrázolása

Meggondolandók:

- Bizonyos műveletek egyes ábrázolásoknál kézenfekvőek, egyszerűen megvalósíthatók, mások pedig nem.
- Érdemes-e minden műveletet minden ábrázolásra megírni?
  - Ha nem: A gráfábrázoláshoz megvalósított műveletek alapján kiderül, hogy egyes gráf-algoritmusok melyik változata készíthető el.
  - Ha igen: Bármely ábrázolásra bármely gráf-algoritmus változat elkészíthető. Közülük – ha szükséges – hatékonysági szempontok alapján választhatunk.

Gráfok ábrázolása

Felépítés: éllistából csúcsmátrix:

```
Típus
   Éllista=Tömb(1..Maxél:TÉl)
   TÉl=Tömb(1..2:TPont)
   Csúcsmátrix=Tömb(1..Maxpont,1..Maxpont:Logikai)
Változó Élszám,Pontszám: Egész
        Cs: Csúcsmátrix
        Fok,Befok,Kifok: Tömb(1..Maxpont,Egész)
```

Gráfok ábrázolása

Felépítés: éllistából csúcsmátrix:

```
Felépít(E,Cs,Fok):
  Cs:=(hamis,…,hamis); Fok():=(0,…,0)
  Ciklus i=1-től Élszám-ig
    Be: x,y
    Cs(x,y):=igaz; Fok(x):=Fok(x)+1
    Cs(y,x):=igaz; Fok(y):=Fok(y)+1
  Ciklus vége
Eljárás vége.
```

Gráfok ábrázolása

Felépítés: éllistából csúcslista (tömb):

```
Típus
  Éllista=Tömb(1..Maxél:TÉl)
  TÉl=Tömb(1..2:TPont)
  Csúcslista=Tömb(1..Maxpont:
                           Tömb(1..Maxpont:TPont)))
Változó Élszám,Pontszám: Egész
        Ki: Csúcslista
         Fok,Befok,Kifok: Tömb(1..Maxpont,Egész)
         E: Éllista
```

Megjegyzés: Dinamikus tömbök esetén a piros Maxpont nem szükséges.

Gráfok ábrázolása

Felépítés: éllistából csúcslista (tömb):

```
Felépít(E,Ki,Fok):
  Fok():=(0,…,0)
  Ciklus i=1-től Élszám-ig
    Be: x,y
    Fok(x):=Fok(x)+1; Ki(x).él(Fok(x)):=y
    Fok(y):=Fok(y)+1; Ki(y).él(Fok(y)):=x
  Ciklus vége
Eljárás vége.
```

Gráfok műveletei

Műveletek csúcsmátrixra:

```
Vanél?(Cs,p,q):
  Vanél?:=Cs(p,q)
Függvény vége.
Élhossz(Cs,p,q):
  Élhossz:=Cs(p,q)
Függvény vége.
```

A többi művelet nehéz, csúcsmátrix esetén nem valósítjuk meg.

Gráfok műveletei

Műveletek csúcslistára (tömb):

```
Szomszédpontokszáma(Ki,p):
  Szomszédpontokszáma:=Fok(p) {vagy Kifok(p)}
Függvény vége.
Szomszéd(Ki,p,i):
  Szomszéd:=Ki(p,i)
Függvény vége.
```

A többi művelet nehéz, csúcslista esetén nem valósítjuk meg.

Gráfok bejárása

A gráf bejárása = minden elem feldolgozása

Probléma:

- Lineáris elrendezésű sokaság (sorozat) bejárása könnyű, egyetlen ciklussal elvégezhető.
- Hálós struktúra bejárása nem kézenfekvő, többféle stratégiával végezhető.

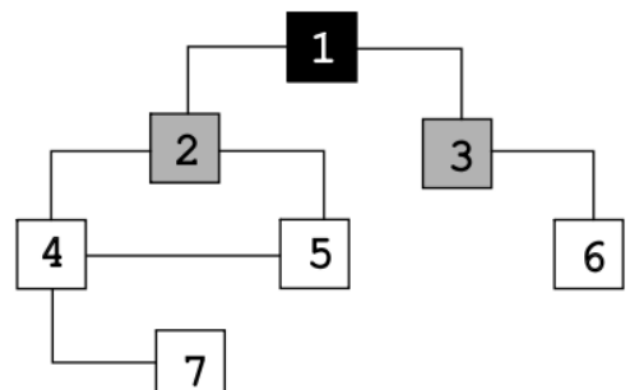
Gráfok bejárása

Gráfbejárás:

- kiindulunk egy tetszőleges pontból,
- éleken haladva eljutunk az összes ponthoz.

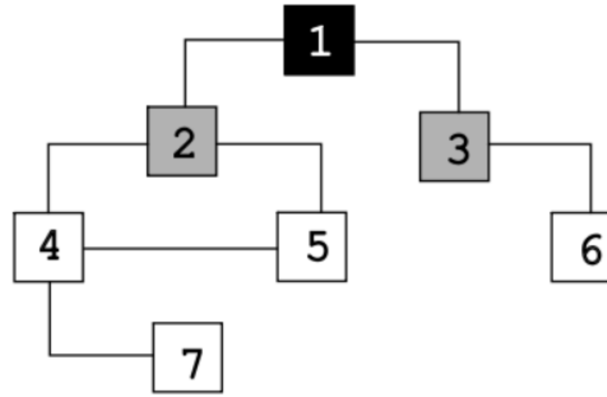
Demonstrálás színekkel:

- Fehér pontok: ahova még nem jutottunk el.
- Szürkék: ahova már eljutottunk, de még „dolog van vele”.
- Feketék: ahova már eljutottunk, s minden belőlük kivezető élt is megvizsgáltunk.



Gráfok bejárása

Demonstrálás színekkel:



- A gráfbejárás kiinduló állapotában egyetlen pont szürke, az összes többi pedig fehér.
- A végállapotban minden pont fekete (ha elérhető a kezdőpontból).

A színekkel tehát a pontok halmazát három részhalmazra bontottuk.

A gráfbejárás pontokat sorol át egyik részhalmazból egy másik részhalmazba (fehérből szürkébe, szürkéből feketébe).

Ehhez a szürkéket vizsgálja!

Gráfok bejárása

Két alapvető stratégia:

- *Szélességi bejárás*: a szürke színűek közül abból lépjünk tovább, amelyikbe legrégebben léptünk.
- *Mélységi bejárás:* abból a pontból lépünk mindig tovább, amelyik legkésőbb került a szürke színűek közé.

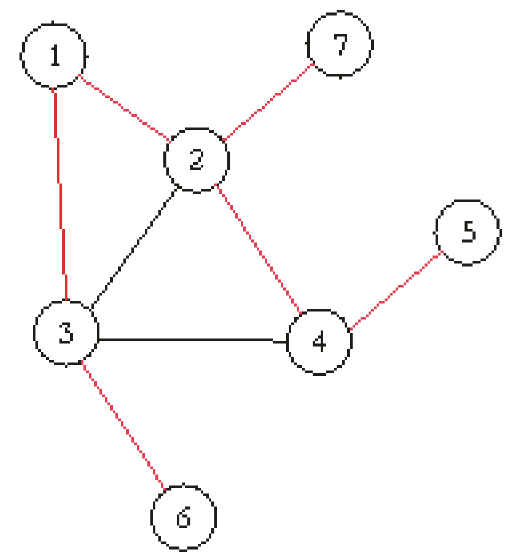
Mindkét bejárásban a szürkék keletkezési sorrendjét kell követnünk valamilyen módon.

Lehetnek (lesznek) további stratégiák is.

Szélességi bejárás

Szélességi bejárás:

- Adatszerkezet, amiből a legrégebben bekerült lép ki először – sor .
- Tároljuk a szürke pontokat egy sorban!
- Van még feldolgozatlan pont = van még szürke pont = nem üres a sor!
- A bejárás egy feszítőfát hoz létre (szélességi feszítőfa).
- Van kör = van nem piros színű él, azaz szürke pontba vezető.



Sorrend: 1,2,3,4,7,6,5

Szélességi bejárás

```
Szélességi bejárás(p):
  Szín(p):=szürke; Sorba(p)
  Ciklus amíg nem üresSor?
    Sorból(p); Szín(p):=fekete
    Ciklus i=1-től Szomszédpontokszáma(p)-ig
      j:=Szomszéd(p,i)
      Ha Szín(j)=fehér
         akkor Sorba(j); Szín(j):=szürke
    Ciklus vége
  Ciklus vége
Eljárás vége.
```

Bejárás csúcslista esetén.

Futási idő: O(Élszám)

Szélességi bejárás

```
Szélességi bejárás(p):
  Szín(p):=szürke; Sorba(p); Honnan(p):=p
  Ciklus amíg nem üresSor?
    Sorból(p); Szín(p):=fekete
    Ciklus i=1-től Szomszédpontokszáma(p)-ig
      j:=Szomszéd(p,i)
      Ha Szín(j)=fehér
         akkor Sorba(j); Szín(j):=szürke
               Honnan(j):=p
    Ciklus vége
  Ciklus vége
Eljárás vége.
```

Bejárás csúcslista esetén.

Futási idő: O(Élszám)

Szélességi bejárás

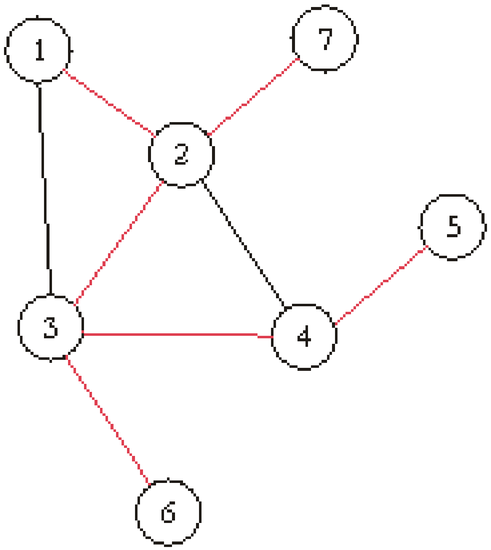
```
Szélességi bejárás(p):
  Szín(p):=szürke; Sorba(p); Táv(p):=0
  Ciklus amíg nem üresSor?
    Sorból(p); Szín(p):=fekete
    Ciklus i=1-től Szomszédpontokszáma(p)-ig
      j:=Szomszéd(p,i)
      Ha Szín(j)=fehér
         akkor Sorba(j); Szín(j):=szürke
               Táv(j):=Táv(p)+1
    Ciklus vége
  Ciklus vége
Eljárás vége.
```

Bejárás csúcslista esetén.

Futási idő: O(Élszám)

Mélységi bejárás:

- Adatszerkezet, amiből a legutoljára bekerült lép ki először – verem.
- Tároljuk a szürke pontokat egy veremben!
- Van még feldolgozatlan pont = van még szürke pont = nem üres a verem!
- A bejárás egy feszítőfát hoz létre (mélységi feszítőfa).
- A verem megtakarítható rekurzióval (hiszen azt veremmel valósítjuk meg).



Sorrend: 1,2,3,4,5,6,7

Mélységi bejárás

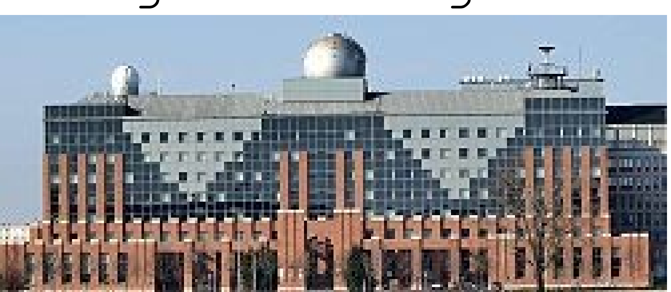
```
Mélységi bejárás(p):
  Szín(p):=szürke
  Ciklus i=1-től Szomszédpontokszáma(p)-ig
    j:=Szomszéd(p,i)
    Ha Szín(j)=fehér akkor Mélységi bejárás(j)
  Ciklus vége
  Szín(p):=fekete
Eljárás vége.
Mélységi bejárás indítása(p):
  Szín:=(fehér,…,fehér); Mélységi bejárás(p)
Eljárás vége.
```

Bejárás csúcslista esetén.

Futási idő: O(Élszám)

Mélységi bejárás

```
Mélységi bejárás(p):
  Szín(p):=szürke
  Ciklus i=1-től Szomszédpontokszáma(p)-ig
    j:=Szomszéd(p,i)
    Ha Szín(j)=fehér akkor Honnan(j):=p
                           Mélységi bejárás(j)
  Ciklus vége
  Szín(p):=fekete
Eljárás vége.
Mélységi bejárás indítása(p):
  Szín:=(fehér,…,fehér); Mélységi bejárás(p)
  Honnan(p):=p
Eljárás vége.
```

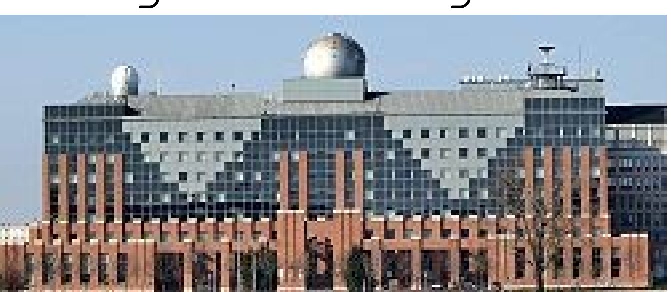


Bejárás csúcslista esetén.

Futási idő: O(Élszám)

Mélységi bejárás

```
Mélységi bejárás(p):
  Szín(p):=szürke
  Ciklus i=1-től Szomszédpontokszáma(p)-ig
    j:=Szomszéd(p,i)
    Ha Szín(j)=fehér akkor Táv(j):=Táv(p)+1
                           Mélységi bejárás(j)
  Ciklus vége
  Szín(p):=fekete
Eljárás vége.
Mélységi bejárás indítása(p):
  Szín:=(fehér,…,fehér); Mélységi bejárás(p)
  Táv(p):=0
Eljárás vége.
```



Bejárás csúcslista esetén.

Futási idő: O(Élszám)

Bejárások alkalmazásai

Összefüggő-e egy irányítatlan gráf?

Alapötlet: bejárás…, s ha *minden* pontba eljutottunk a bejárás során, akkor *összefüggő*.

```
Összefüggő?:
  Szín(1..Pontszám):=fehér; Bejárás(1)
  i:=1
  Ciklus amíg i≤Pontszám és Szín(i)=fekete
    i:=i+1
  Ciklus vége
  Összefüggő?:=(i>Pontszám)
Függvény vége.
```

Kérdés: irányított gráf összefüggő vagy erősen összefüggő?

Bejárások alkalmazásai

Irányítatlan gráf összefüggő komponensei száma

Alapötlet: bejárás újraindul minden fehéren maradt pontból.

```
Komponensek száma:
  Szín(1..Pontszám):=fehér; Db:=0
  Ciklus i=1-től Pontszám-ig
     Ha Szín(i)=fehér akkor Db:=Db+1; Bejárás(i)
  Ciklus vége
  Komponensek száma:=Db
Függvény vége.
```

Bejárások alkalmazásai

Irányítatlan gráf összefüggő komponensei

Alapötlet: bejárás újraindul minden fehéren maradt pontból, feketévé válásnál komponens sorszámot tárol.

```
Bejárás(p,Db):
   …
   Szín(p):=fekete; Komponens(p):=Db
   …
Eljárás vége.
```

Van-e út 2 pont között

Alapötlet: Az egyik pontból indul a bejárás, ha a másik fekete lesz, akkor van út.

```
Van út?(p,q):
  Szín(1..Pontszám):=fehér
  Bejárás(p)
  Van út?:=(Szín(q)≠fehér)
Függvény vége.
```

Ötlet: Ha a másik pont szürkévé vált, akkor már tudjuk, hogy van út, befejezhetjük a bejárást:

```
  Ciklus amíg nem Üres…? és Szín(q)=fehér
```

Kérdés – miben más: Van-e kör egy p ponton keresztül?

Bejárások alkalmazásai

Út megadása 2 pont között

Alapötlet: Az egyikből indul a bejárás, ha a másik fekete lesz, akkor van út. Ekkor induljunk visszafelé a célpontból!

```
Út(p,q):
  Szín(1..Pontszám):=fehér; Bejárás(p)
  Ha Szín(q)=fekete akkor Útkiírás(q,p)
Függvény vége.
Útkiírás(q,p):
  Ha q≠p akkor Útkiírás(Honnan(q),p)
  Ki: q
Eljárás vége.
```

Megjegyzés: Minden elért (azaz fekete) pontból bejárhatjuk azt az utat, amin eljutottunk oda.

Gráfok
1. előadás vége