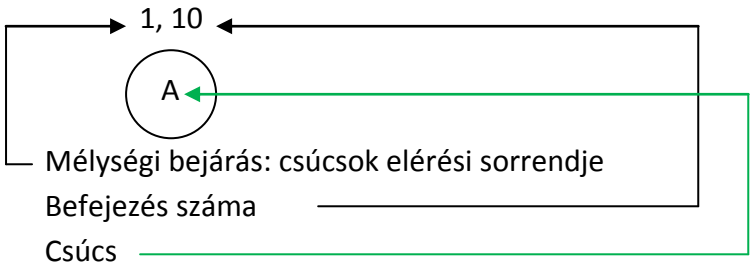
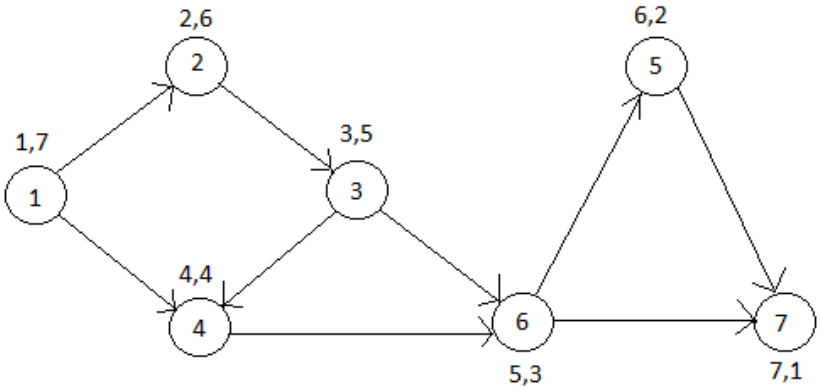
Algoritmusok és adatszerkezetek 2.

Varga Balázs gyakorlata alapján
Készítette: Nagy Krisztián

10. gyakorlat

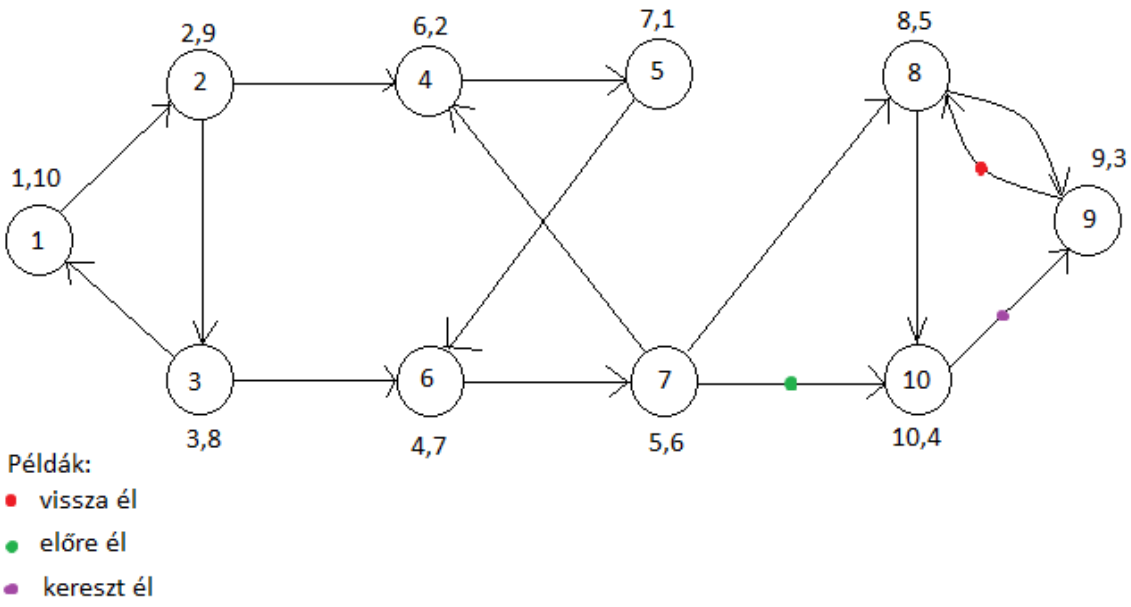
Mélységi bejárás





Megbeszélés alapján mindig a legkisebb csúcs irányába haladunk.

Topológikus rendezés: Befejezési számok csökkenő sorrendje alapján a csúcsok felírása.
1,2,3,4,6,5,7
Megjegyzés: Visszaél-et nem tartalmazhat. Amennyiben visszaél szerepelne az ábránkon, úgy nem lehetne topológikusan rendezni a gráf csúcsait.



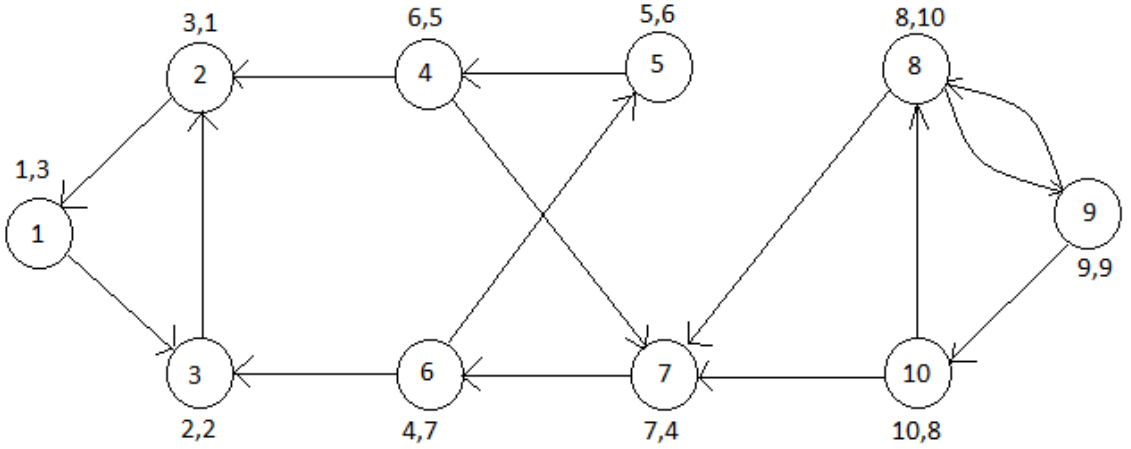
A visszaélek miatt nem rendezhetőtopológikusan!

Viszont a komponensek egyszerűbb meghatározásának érdekében írjuk fel a csúcsokat úgy, mint ha topológikusan „lenne” rendezve. (Ismételten, amennyiben az a kérdés, hogy topológikusan rendezhető-e a fentebbi gráf a helyes válasz a NEM!)

1,2,3,6,7,8,10,9,4,5

Rajzoljuk fel a Transzponált gráfot és indítsuk el rajta újra a mélységi bejárást. Ahol elakadunk megnézzük, hogy a fentebb leírt számsorból melyik csúcsokat jártuk be és a kimaradt/ rákövetkező csúcson újra elindítjuk a mélységi bejárást folytatva az első bejárás számozását. Minden egyes újraindított mélységi bejárás azt jelenti, hogy a gráfunkban egy új komponenst találtunk!

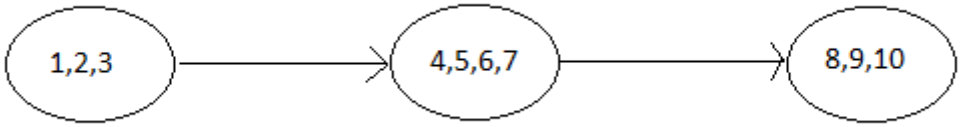
Transzponált gráf: Az eredeti gráf az élek irányításának a megfordításával.



1,2,3,6,7,8,10,9,4,5 volt a fentebb megalkotott sorozat.
1,3,2 – elakadtunk – 6-tól folytatjuk 6,5,4,7 – elakadtunk – 8-tól folytatjuk – 8,9,10

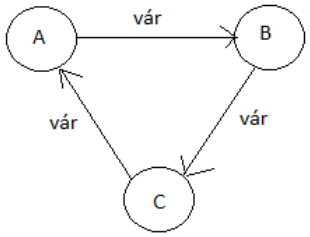
Írjuk fel a redukált gráfot!

Redukált gráf: A gráf csúcsai az egyes komponensek csúcsai. Két csúcs közötti irányítás pedig az **eredeti** gráf irányítása alapján. Amennyiben kört találunk a redukált gráfban, úgy valamit elszúrtunk, mivel ekkor az egész gráfunk egy komponensű lenne.



<u>Gyakorlati haszna:</u>

Például operációs rendszerek esetén a Holtpont fogalma: 3 egymástól függő futó folyamat egymásra vár. A fentebbi módszerrel fel lehet fedni ezt a problémát. Ekkor az operációs rendszer általában kilövi az egyik folyamatot.



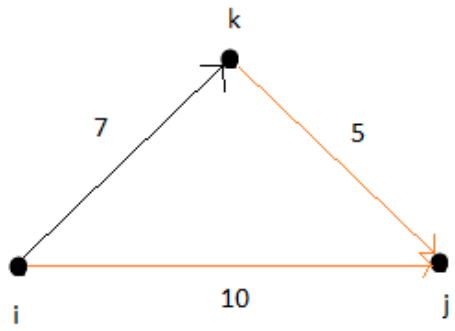
Ekkor például a C-vel jelölt folyamatot megszüntetve / leállítva megoldódik a probléma.

Floyd-algoritmus

Használata: Egy gráf minden csúcsából, minden csúcsába megadja a legrövidebb utat n^3-ös hatékonysággal.

$D^{(k)}[i,j]$, ahol:
- (k)- k val $i \sim j$ legrövidebb út és a közbülső csúcsok maximális indexe k.
- $i \to j$ a legrövidebb út hossza.
- Amennyiben $k = n$-hez értünk, akkor készen vagyunk.
- $k = 0$ az init továbbá $k = 1..n$

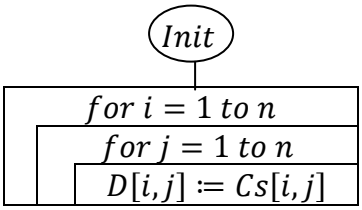
$D^{(k-1)}[i,j] \quad \to \quad D^{(k)}[i,j]$

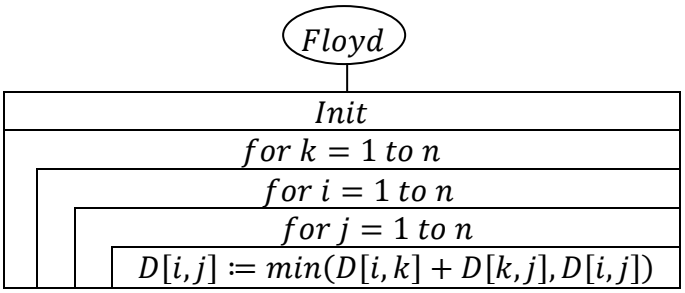


$$min\left(D^{(k-1)}[i,k] + D^{(k-1)}[k,j], D^{(k-1)}[i,j]\right)$$

vegye be k-t — ne vegye be k-t

$D^{(0)}[i,j]$ → Csúcs mátrix
Init lépés: a csúcsmátrix áttöltése D-be





Ha az útvonalat is szeretnénk megállapítani, akkor Dijkstra algoritmust kell futtatnunk minden csúcsra.

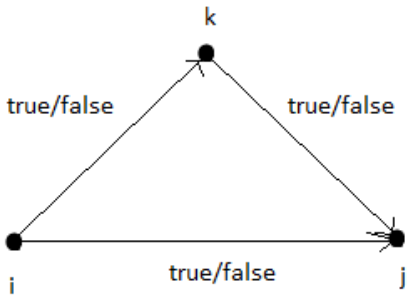
Warshall-algoritmus

Használata: Egy gráf minden csúcsából, minden csúcsába megadja, hogy van-e elérési út n^3-ös hatékonysággal.

Elve úgyan az, mint a Floyd-algoritmusnak

$W^{(k)}[i,j]$ val jelöljük a mátrixunkat.

$W^{(k-1)}[i,j] \quad \rightarrow \quad W^{(k)}[i,j]$



$W[i,j] := W[i,j] \vee (W[i,k] \wedge W[k,j])$