A számítástudomány alapjai

Mélységi keresés és PERT

2024. október 8.

Hol tartunk, és hogyan tovább?

Hol tartunk, és hogyan tovább?

- Általános gráfbejárás, BFS

Hol tartunk, és hogyan tovább?

- Általános gráfbejárás, BFS
- Legrövidebb utak keresése (r, ℓ)-felső becslés élmenti javításával, Dijkstra-algoritmus nemnegatív hosszfüggvényre

Hol tartunk, és hogyan tovább?

- Általános gráfbejárás, BFS
- Legrövidebb utak keresése (r, ℓ)-felső becslés élmenti javításával, Dijkstra-algoritmus nemnegatív hosszfüggvényre
- Konzervatív hosszfüggvény esetén is tudunk legrövidebb utat találni

Hol tartunk, és hogyan tovább?

- Általános gráfbejárás, BFS
- Legrövidebb utak keresése (r, ℓ)-felső becslés élmenti javításával, Dijkstra-algoritmus nemnegatív hosszfüggvényre
- Konzervatív hosszfüggvény esetén is tudunk legrövidebb utat találni
- Példa egy másik fajta gráfbejárásra és alkalmazásaira.

Hol tartunk, és hogyan tovább?

- Általános gráfbejárás, BFS
- Legrövidebb utak keresése (r, ℓ)-felső becslés élmenti javításával, Dijkstra-algoritmus nemnegatív hosszfüggvényre
- Konzervatív hosszfüggvény esetén is tudunk legrövidebb utat találni
- Példa egy másik fajta gráfbejárásra és alkalmazásaira.
- Leghosszabb utak keresése.

Depth First Search

Milyen gráfbejárás lehet a BFS ,,ellentéte”?

Depth First Search

Depth First Search

Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

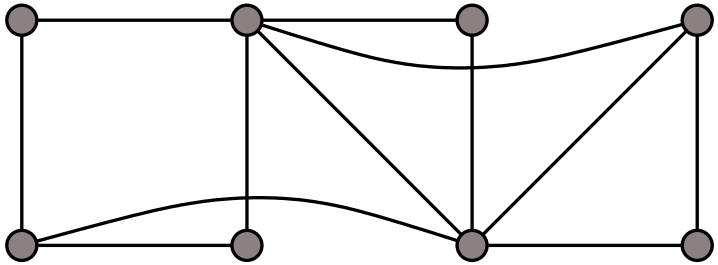
Megj: A DFS olyan bejárás, amikor az $\boxed{1.}$ esetben (amikor van elért csúcs) mindig a legutolsónak elért csúcsot választjuk.

Depth First Search

Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Depth First Search

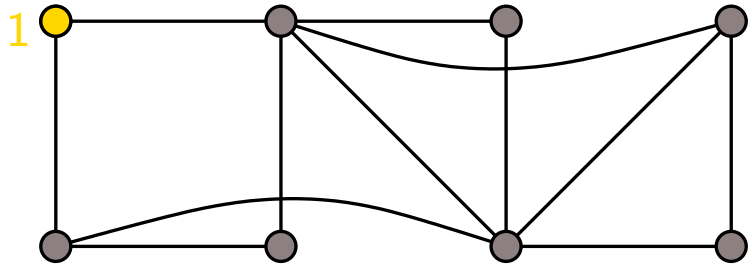


Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

Depth First Search

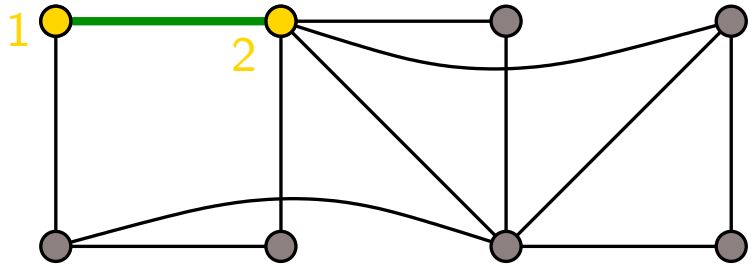


Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

Depth First Search

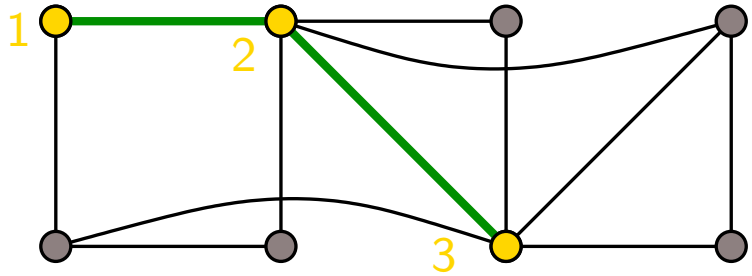


Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

Depth First Search

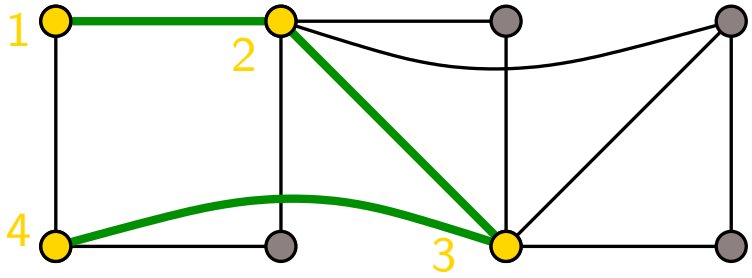


Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

Depth First Search

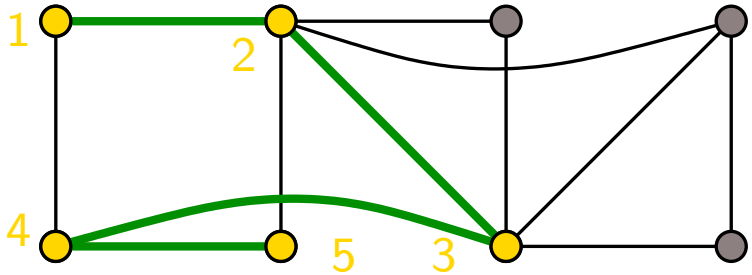


Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

Depth First Search

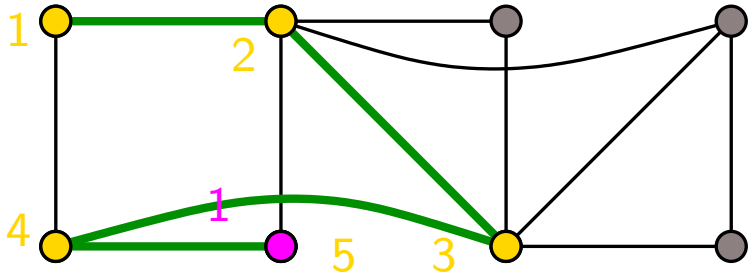


Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

Depth First Search

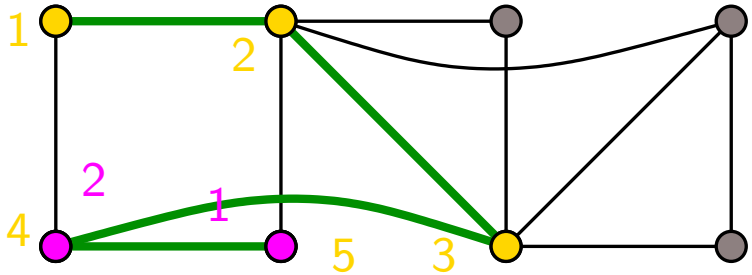


Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

Depth First Search

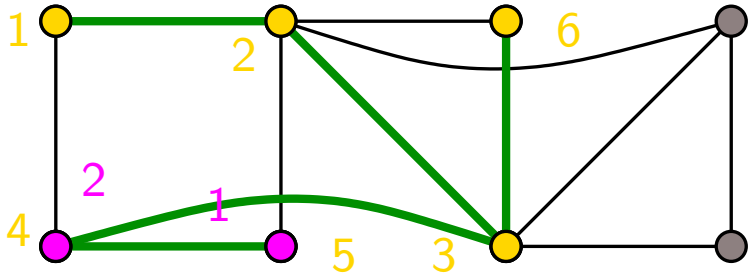


Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

Depth First Search

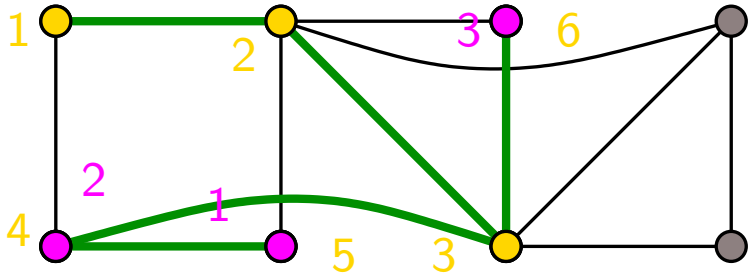


Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

Depth First Search

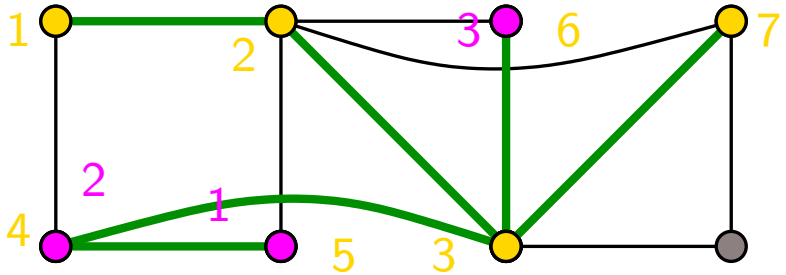


Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

Depth First Search

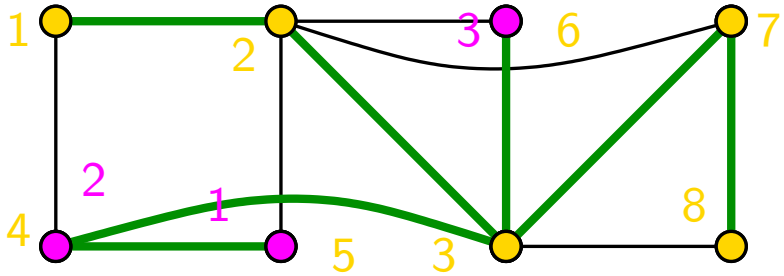


Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

Depth First Search

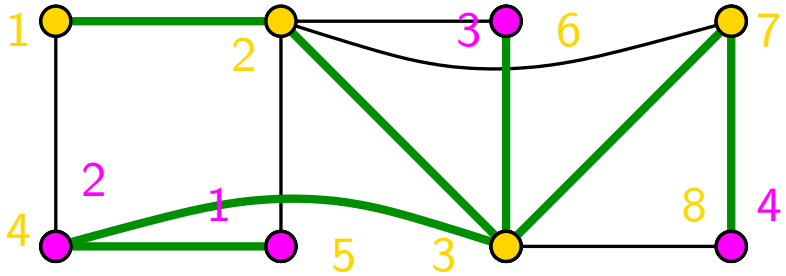


Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

Depth First Search

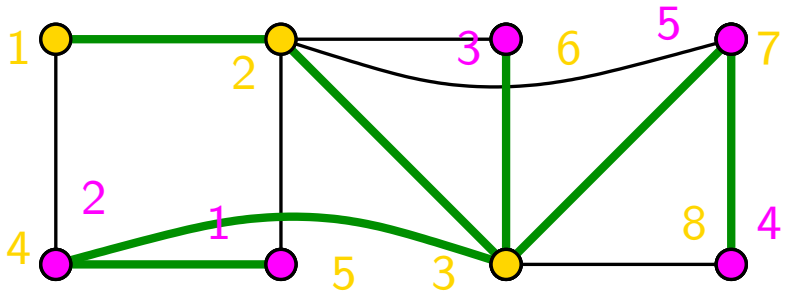


Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

Depth First Search

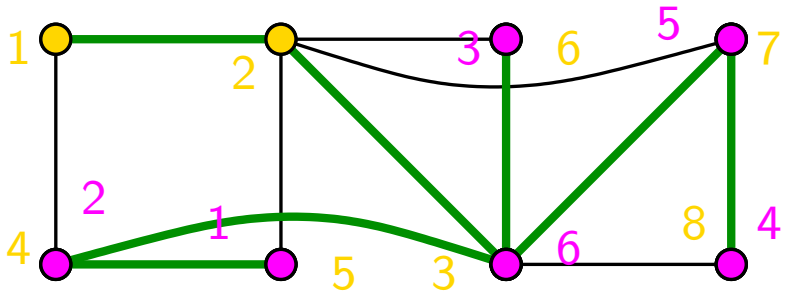


Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

Depth First Search

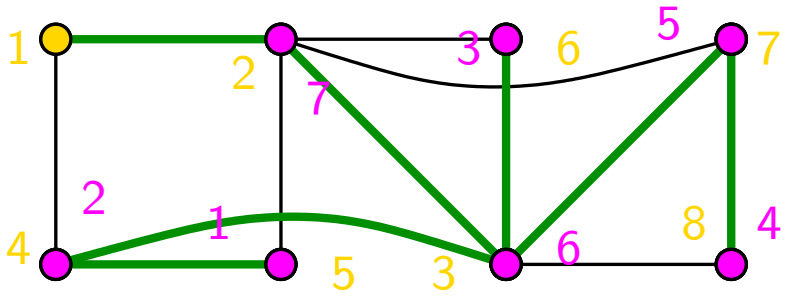


Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

Depth First Search

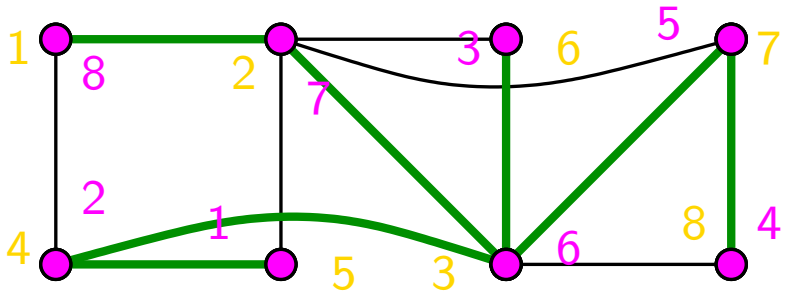


Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

Depth First Search

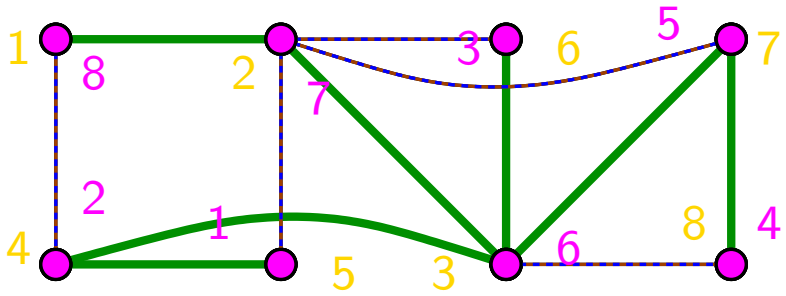


Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

Depth First Search

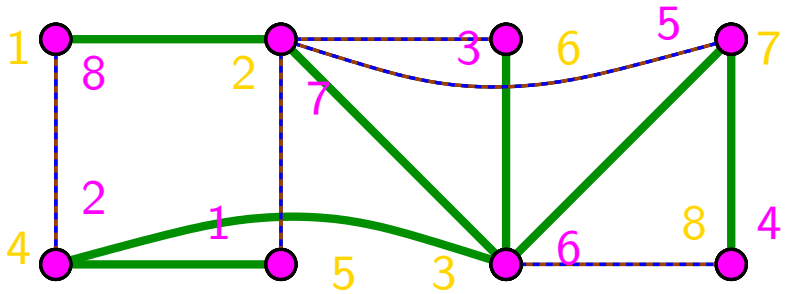


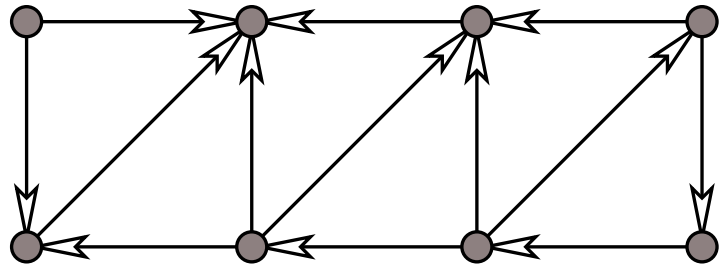
Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

Depth First Search



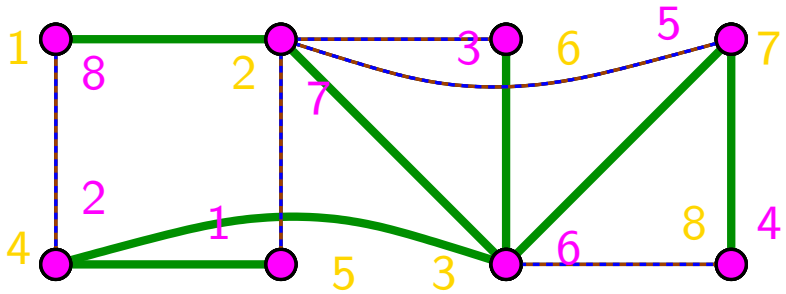


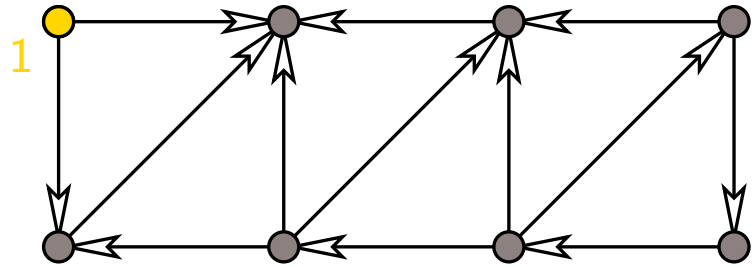
Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

Depth First Search



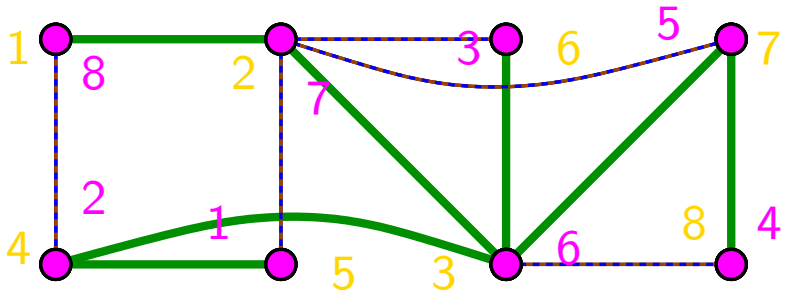


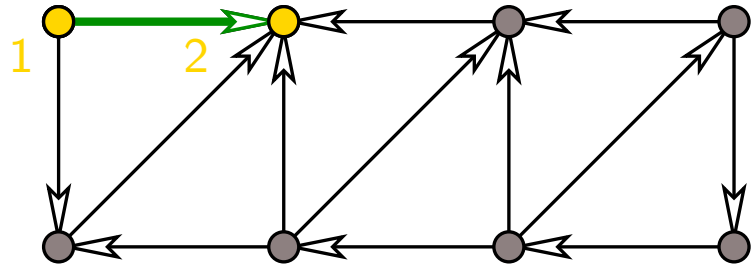
Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

Depth First Search



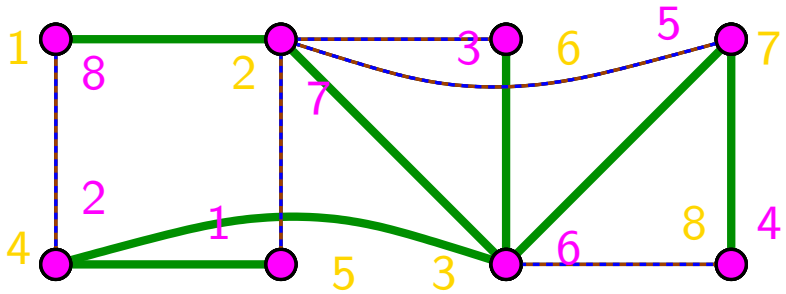


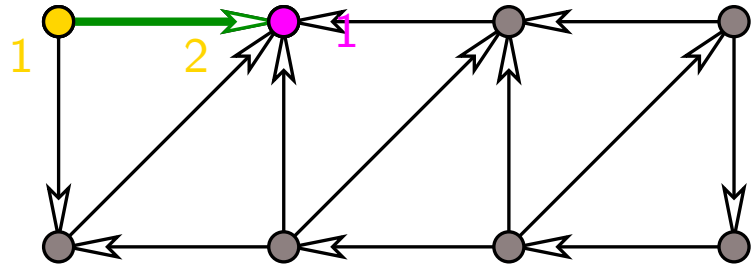
Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

Depth First Search



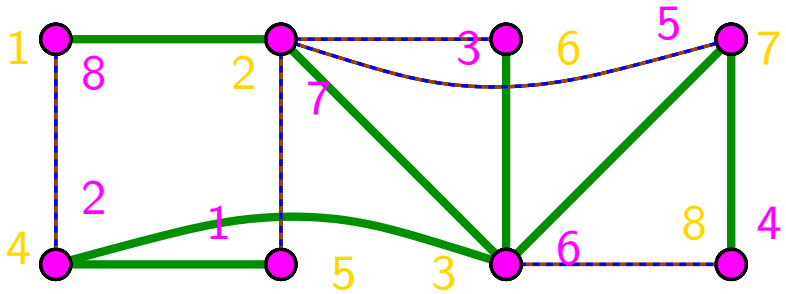


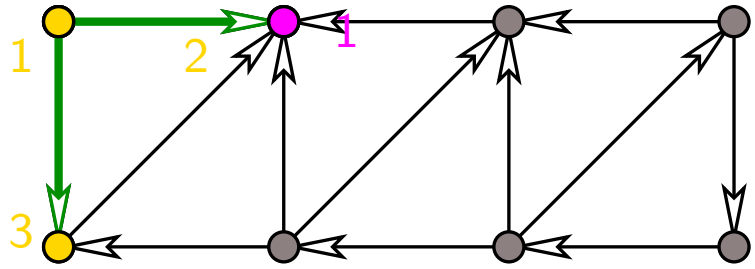
Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

Depth First Search



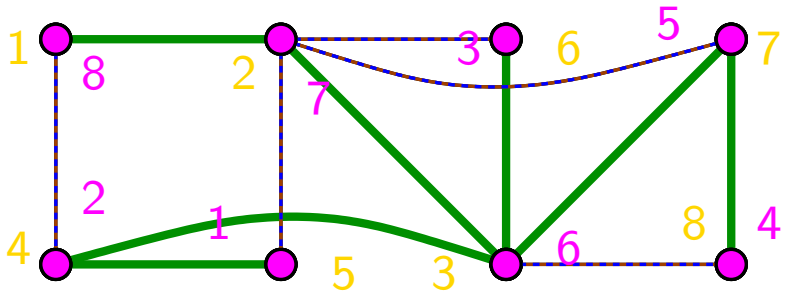


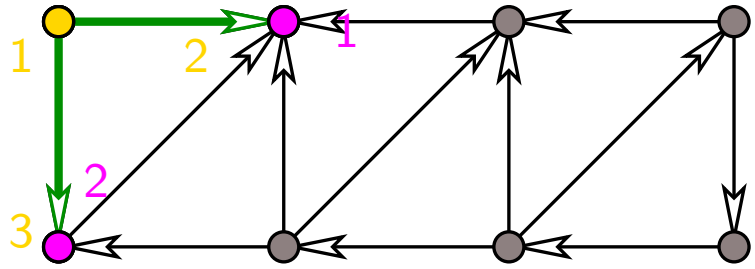
Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

Depth First Search



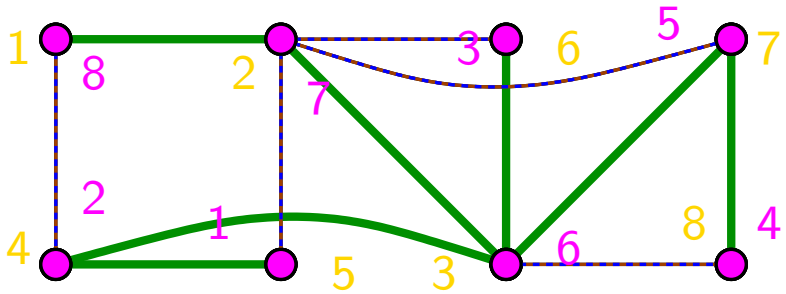


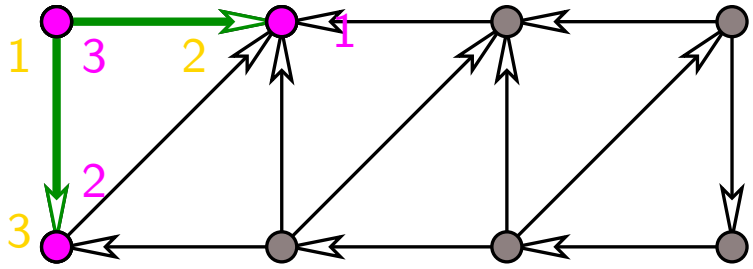
Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

Depth First Search



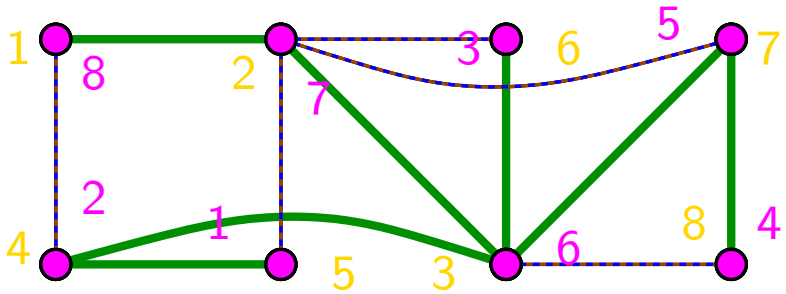


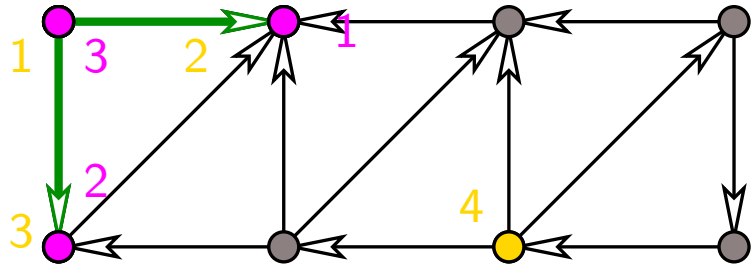
Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

Depth First Search



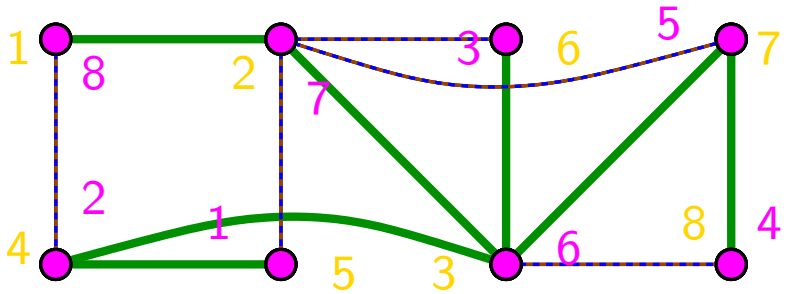


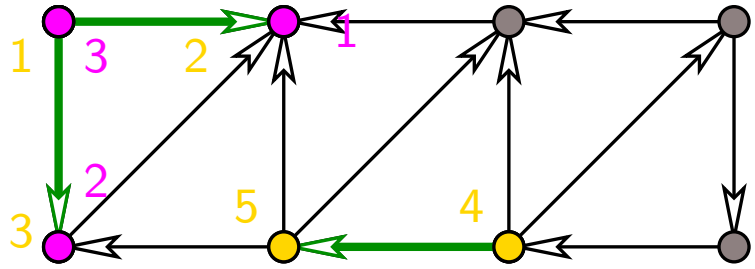
Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

Depth First Search



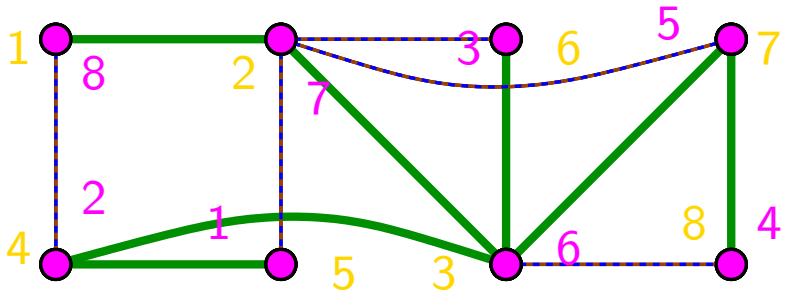


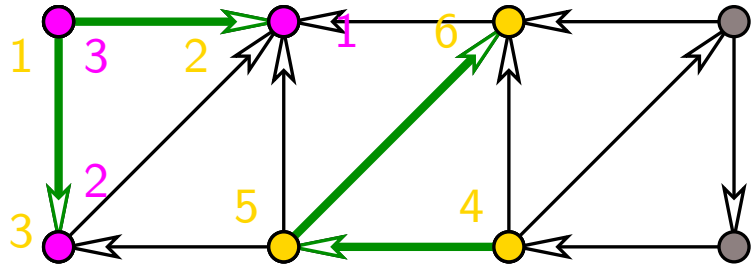
Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

Depth First Search



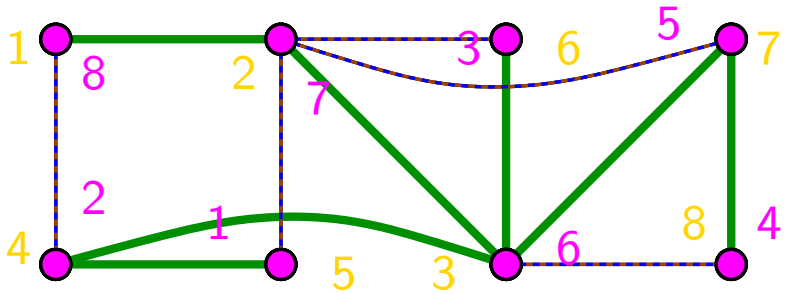


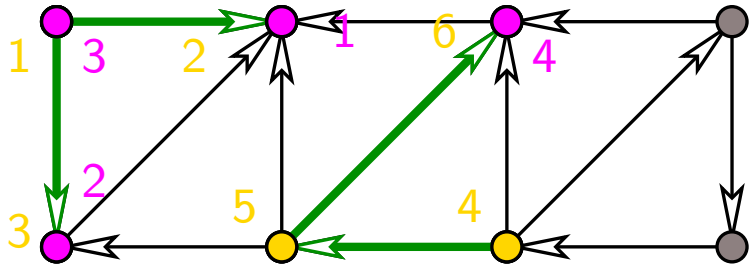
Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

Depth First Search



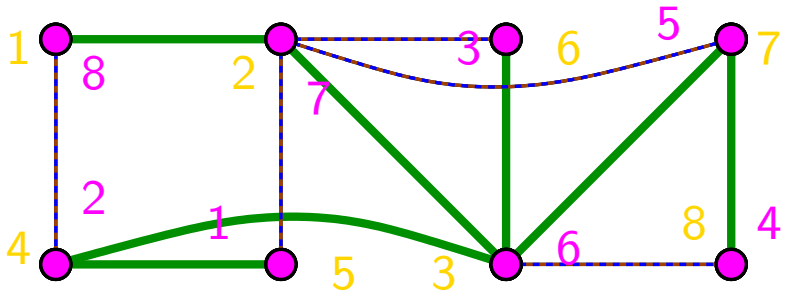


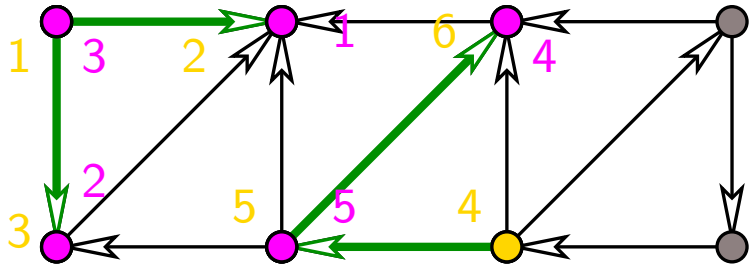
Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

Depth First Search



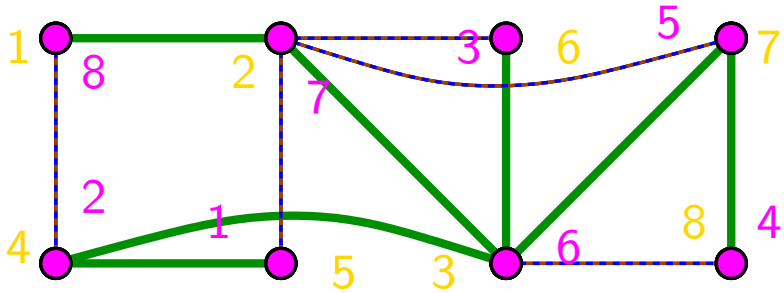


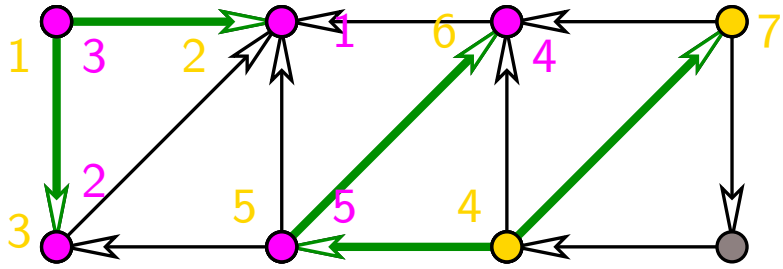
Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

Depth First Search



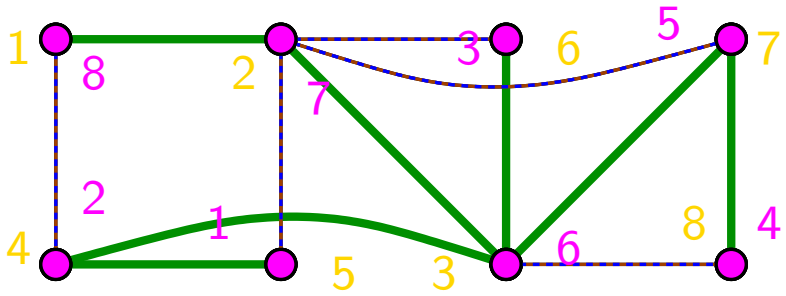


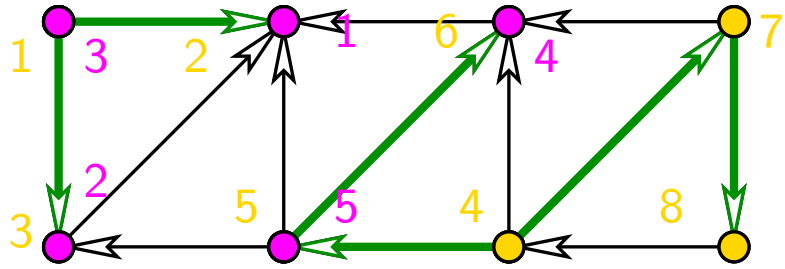
Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

Depth First Search



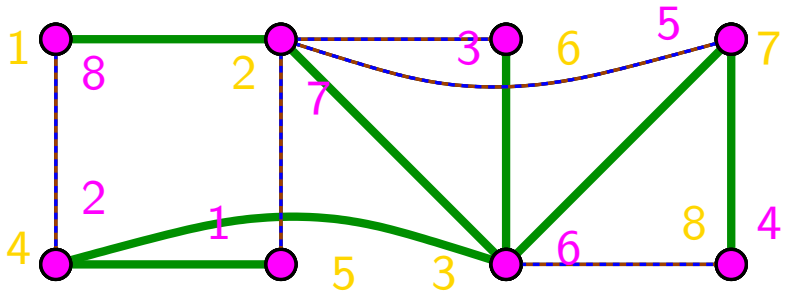


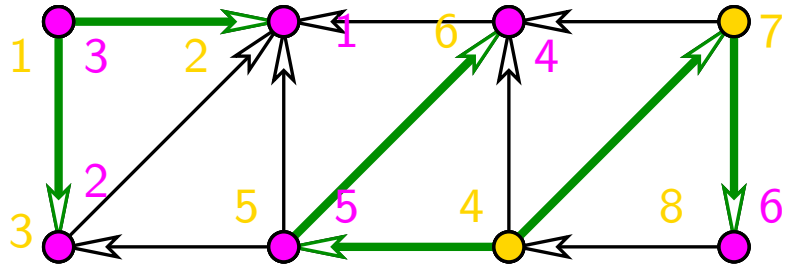
Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

Depth First Search



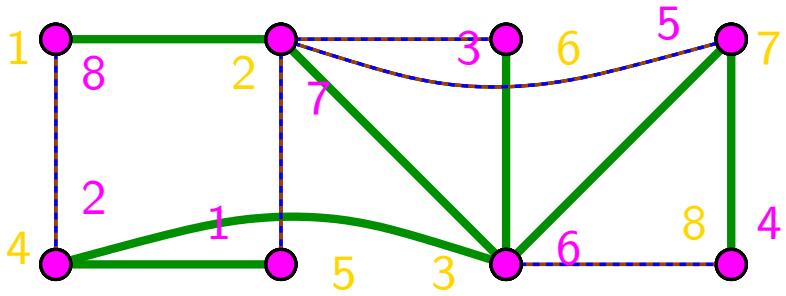


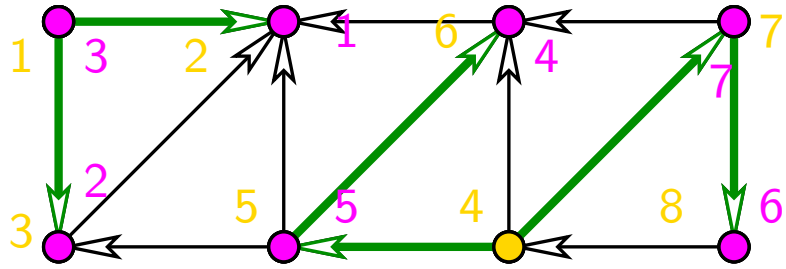
Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

Depth First Search



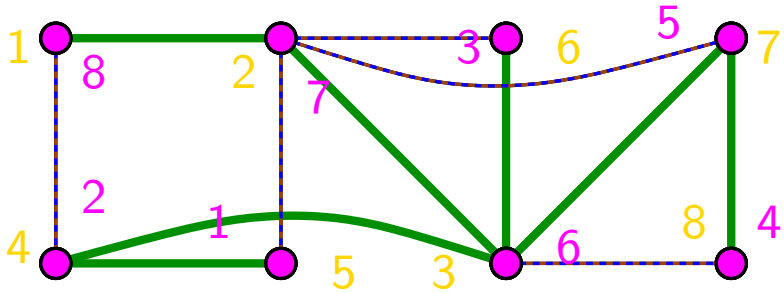


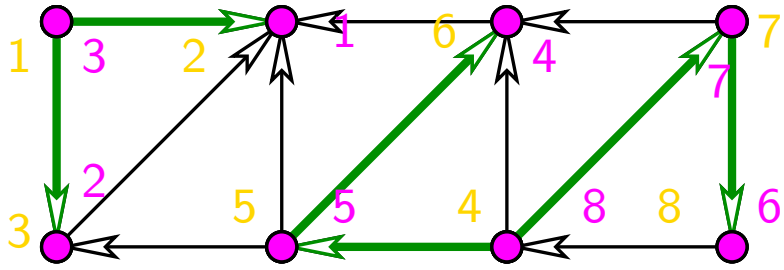
Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

Depth First Search



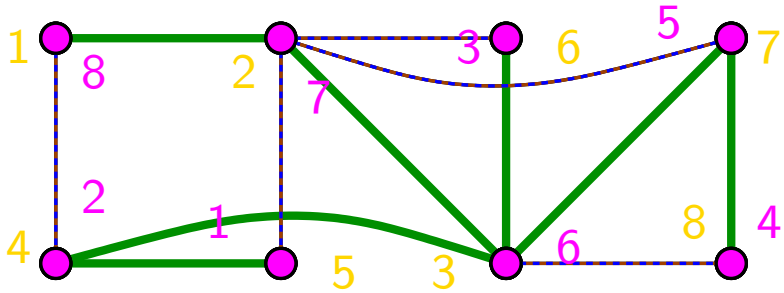


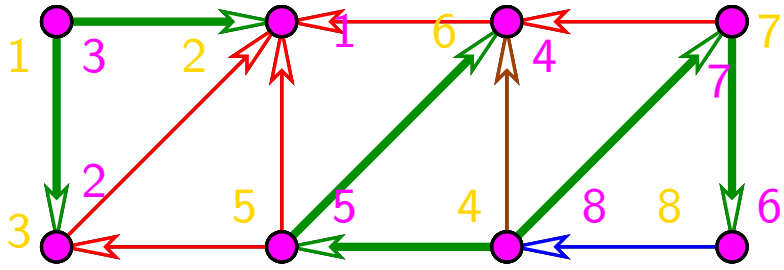
Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

Depth First Search



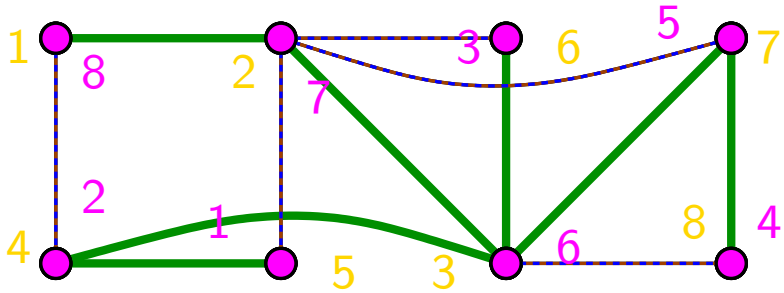


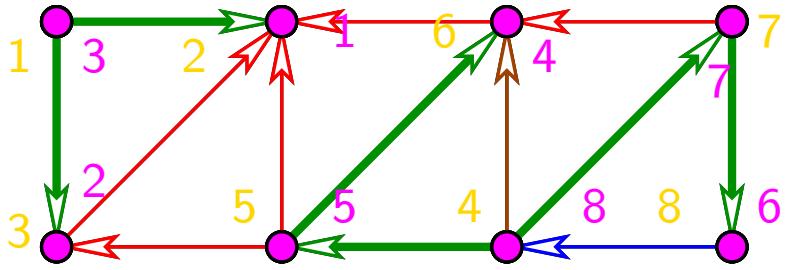
Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

Depth First Search





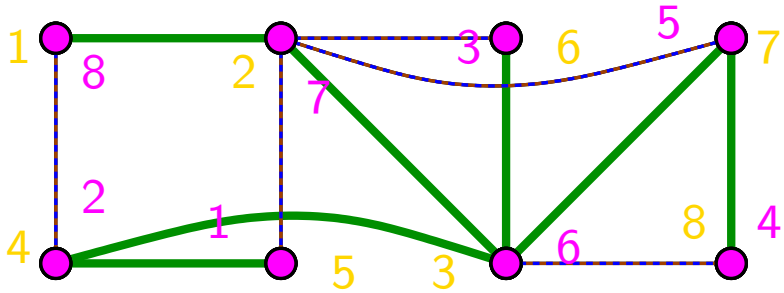
Mélységi bejárás (DFS)

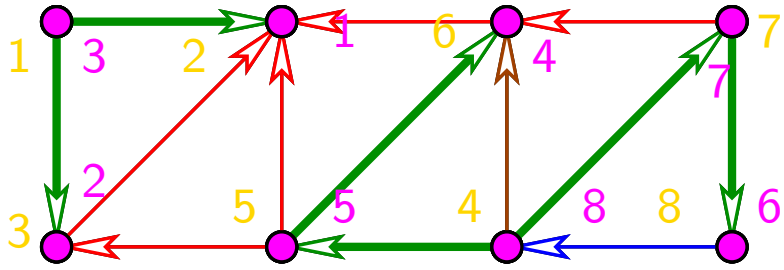
Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

Megj: A BFS konkrét megvalósításában szükség van arra, hogy az elért csúcsokat úgy tároljuk, hogy könnyű legyen kiválasztani az elért csúcsok közül a legkorábban elértet. Erre egy célszerű adatstruktúra a *sor* (avagy *FIFO lista*). Ha a BFS egy konkrét megvalósításában ezt az adatstruktúrát *veremre* (más néven *LIFO listára*) cseréljük, akkor ezzel a DFS-t valósítjuk meg.

Depth First Search



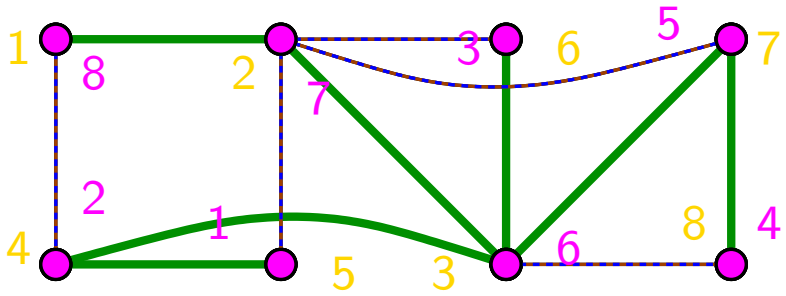


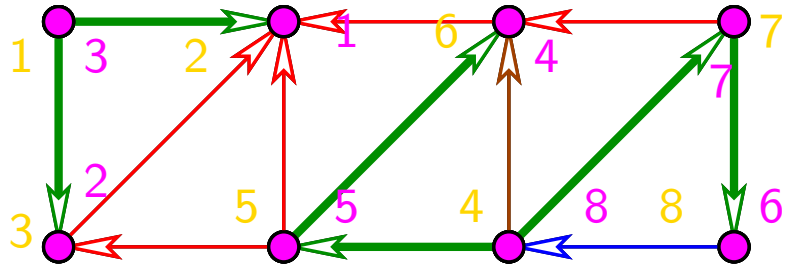
Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

Depth First Search





Mélységi bejárás (DFS)

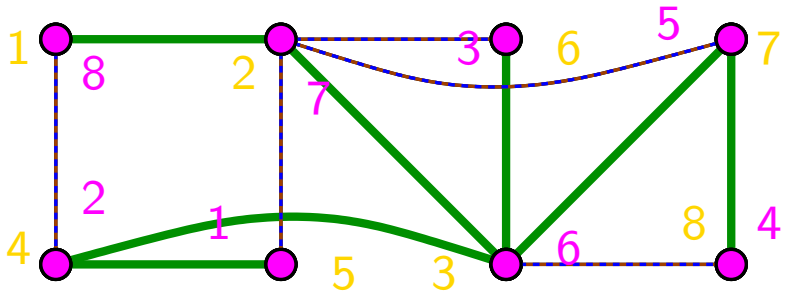
Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

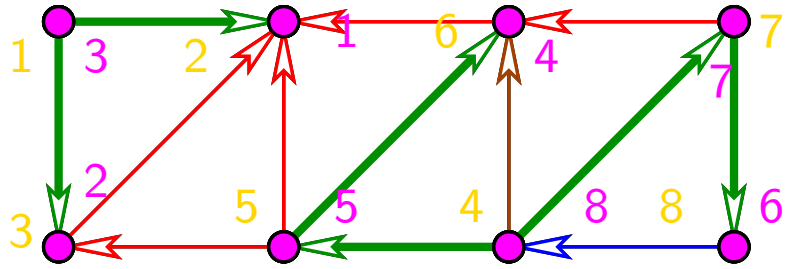
Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

Megf: Tfh a G gráf éleit DFS után osztályoztuk.

(1) Ha uv faél, akkor $m(u) < m(v)$ és $b(u) > b(v)$.

Depth First Search





Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

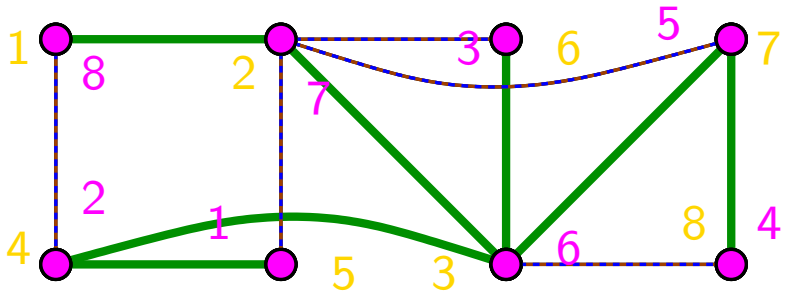
Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

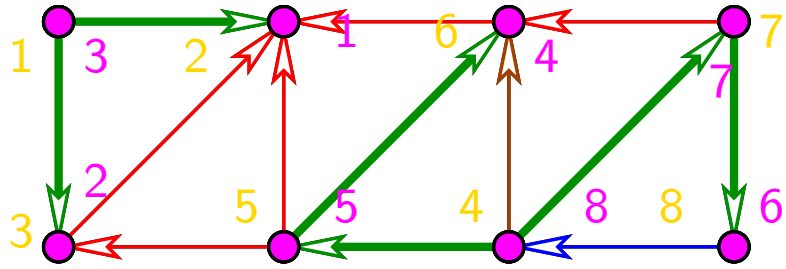
Megf: Tfh a G gráf éleit DFS után osztályoztuk.

(1) Ha uv faél, akkor $m(u) < m(v)$ és $b(u) > b(v)$.

Biz: v-t u-ból értük el, ezért $m(u) < m(v)$. A v elérésekor u és v elért állapotúak. A DFS szabálya szerint mindaddig, amíg v elért állapotú, u-val nem foglalkozunk. Tehát u-t csakis v befejezése után fejezhetjük be, azaz $b(u) > b(v)$. □

Depth First Search





Mélységi bejárás (DFS)

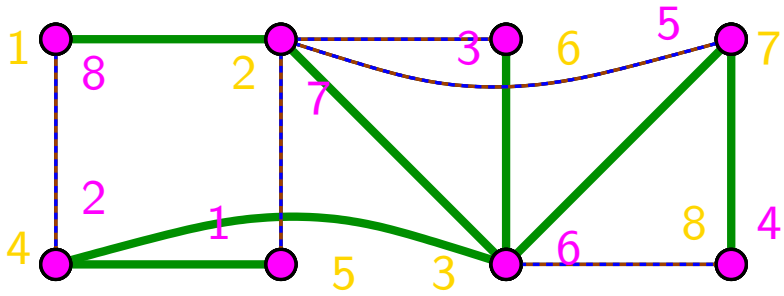
Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

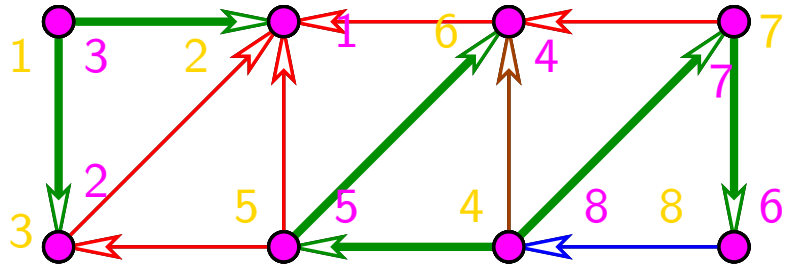
Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

Megf: Tfh a G gráf éleit DFS után osztályoztuk.

(1) Ha uv faél, akkor $m(u) < m(v)$ és $b(u) > b(v)$.

Depth First Search





Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

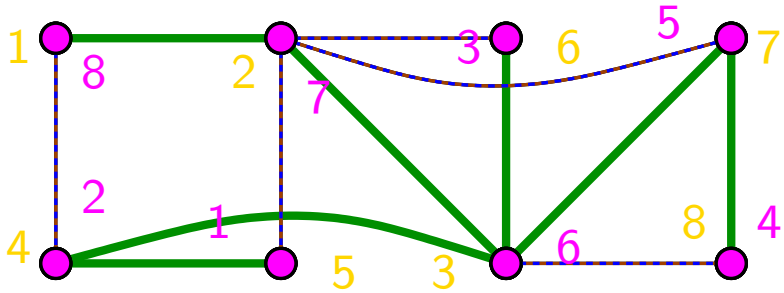
Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

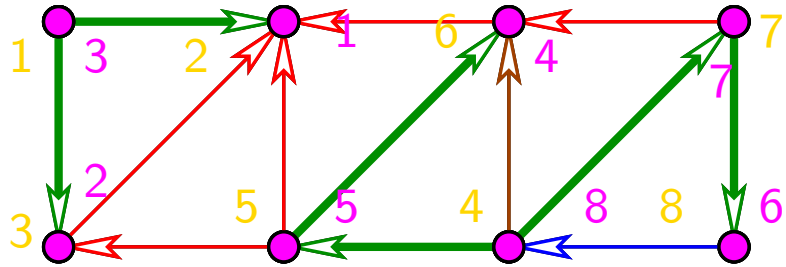
Megf: Tfh a G gráf éleit DFS után osztályoztuk.

(1) Ha uv faél, akkor $m(u) < m(v)$ és $b(u) > b(v)$.

(2) Ha uv előreél, akkor $m(u) < m(v)$ és $b(u) > b(v)$.

Depth First Search





Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

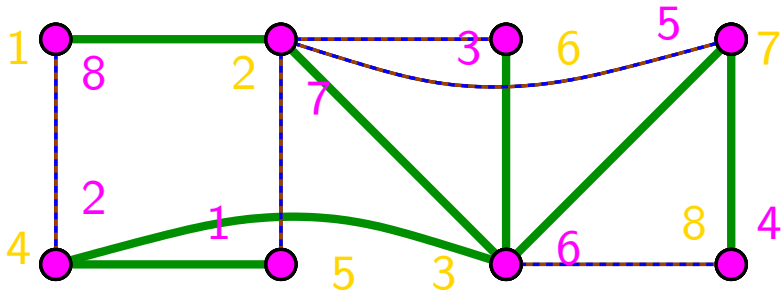
Megf: Tfh a G gráf éleit DFS után osztályoztuk.

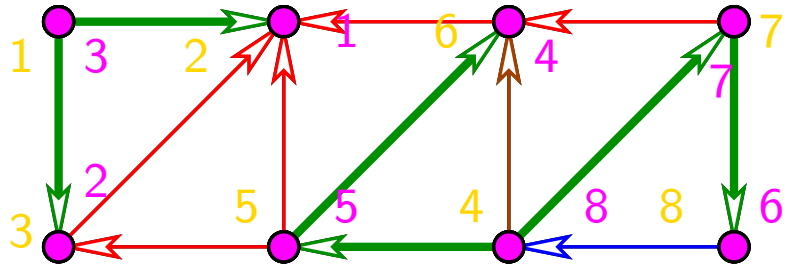
(1) Ha uv faél, akkor $m(u) < m(v)$ és $b(u) > b(v)$.

(2) Ha uv előreél, akkor $m(u) < m(v)$ és $b(u) > b(v)$.

Biz: Mivel uv előreél, ezért u-ból v-be egy faélekből álló irányított út vezet a DFS-fában. (1) miatt a faélek mentén a mélységi szám mindvégig növekszik, a befejezési pedig csökken. □

Depth First Search





Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

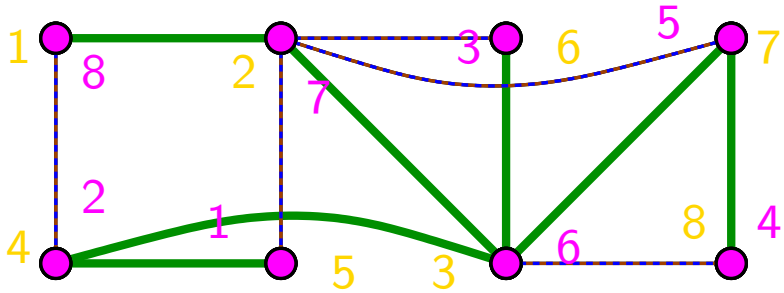
Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

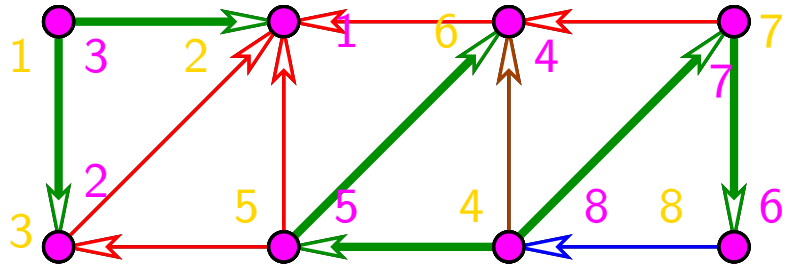
Megf: Tfh a G gráf éleit DFS után osztályoztuk.

(1) Ha uv faél, akkor $m(u) < m(v)$ és $b(u) > b(v)$.

(2) Ha uv előreél, akkor $m(u) < m(v)$ és $b(u) > b(v)$.

Depth First Search





Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

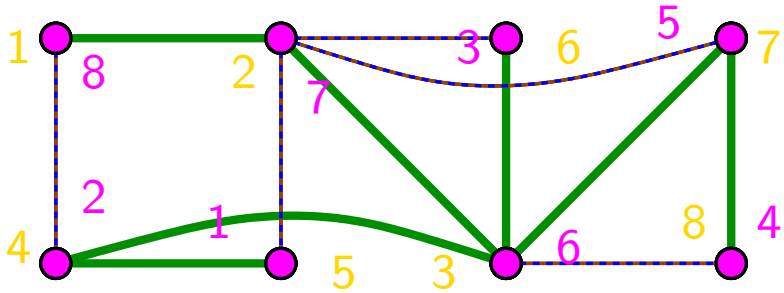
Megf: Tfh a G gráf éleit DFS után osztályoztuk.

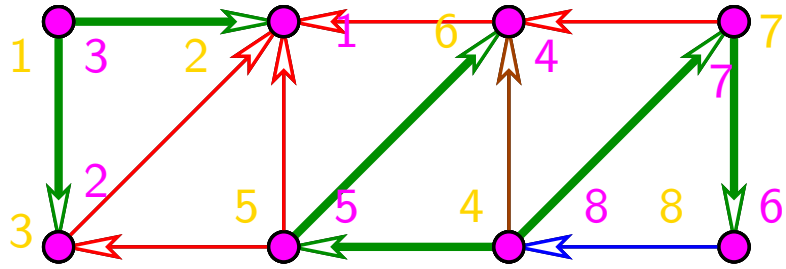
(1) Ha uv faél, akkor $m(u) < m(v)$ és $b(u) > b(v)$.

(2) Ha uv előreél, akkor $m(u) < m(v)$ és $b(u) > b(v)$.

(3) Ha uv visszaél, akkor $m(u) > m(v)$ és $b(u) < b(v)$.

Depth First Search





Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

Megf: Tfh a G gráf éleit DFS után osztályoztuk.

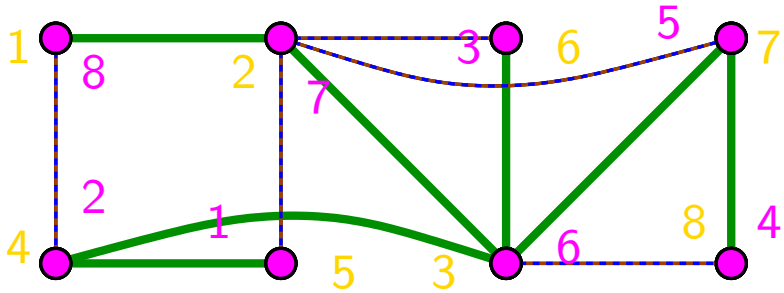
(1) Ha uv faél, akkor $m(u) < m(v)$ és $b(u) > b(v)$.

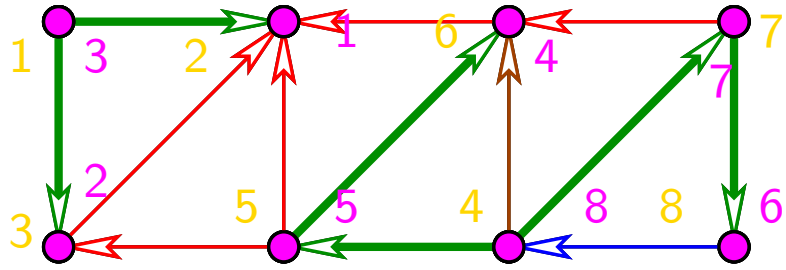
(2) Ha uv előreél, akkor $m(u) < m(v)$ és $b(u) > b(v)$.

(3) Ha uv visszaél, akkor $m(u) > m(v)$ és $b(u) < b(v)$.

Biz: Mivel uv visszaél, ezért v-ből u-ba egy faélekből álló irányított út vezet a DFS-fában. (1) miatt a faélek mentén a mélységi szám mindvégig növekszik, a befejezési pedig csökken. □

Depth First Search





Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

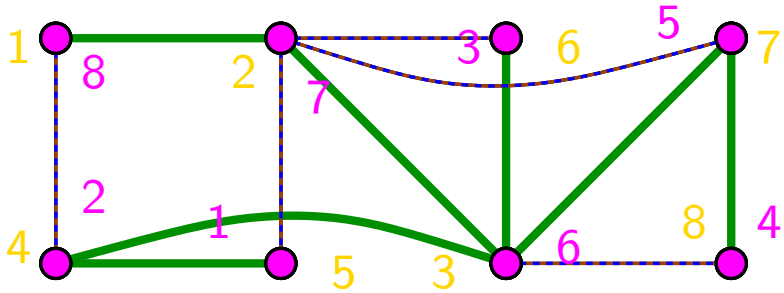
Megf: Tfh a G gráf éleit DFS után osztályoztuk.

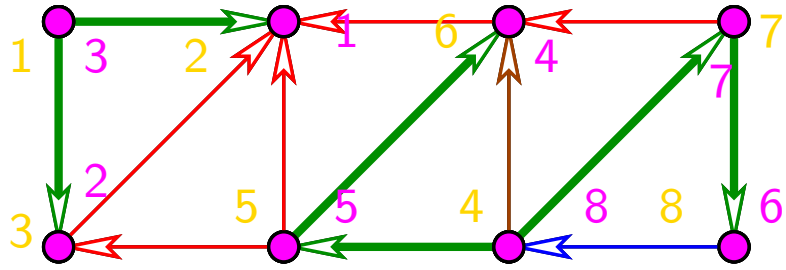
(1) Ha uv faél, akkor $m(u) < m(v)$ és $b(u) > b(v)$.

(2) Ha uv előreél, akkor $m(u) < m(v)$ és $b(u) > b(v)$.

(3) Ha uv visszaél, akkor $m(u) > m(v)$ és $b(u) < b(v)$.

Depth First Search





Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

Megf: Tfh a G gráf éleit DFS után osztályoztuk.

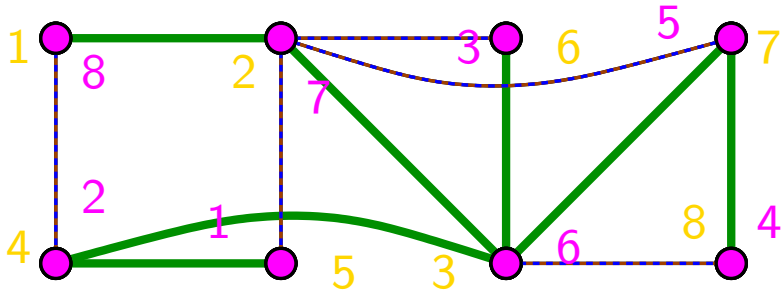
(1) Ha uv faél, akkor $m(u) < m(v)$ és $b(u) > b(v)$.

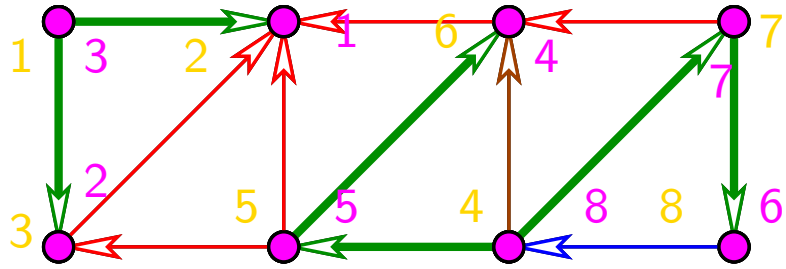
(2) Ha uv előreél, akkor $m(u) < m(v)$ és $b(u) > b(v)$.

(3) Ha uv visszaél, akkor $m(u) > m(v)$ és $b(u) < b(v)$.

(4) Ha uv keresztél, akkor $m(u) > m(v)$ és $b(u) > b(v)$.

Depth First Search





Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

Megf: Tfh a G gráf éleit DFS után osztályoztuk.

(1) Ha uv faél, akkor $m(u) < m(v)$ és $b(u) > b(v)$.

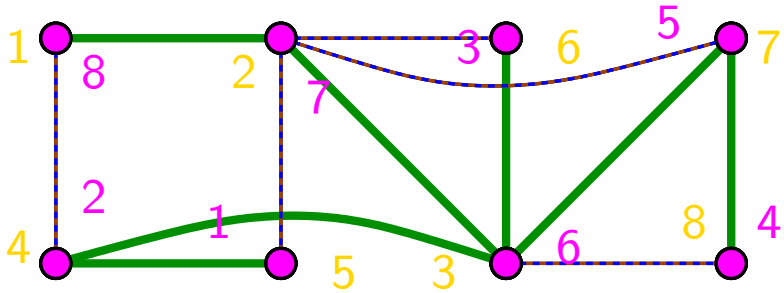
(2) Ha uv előreél, akkor $m(u) < m(v)$ és $b(u) > b(v)$.

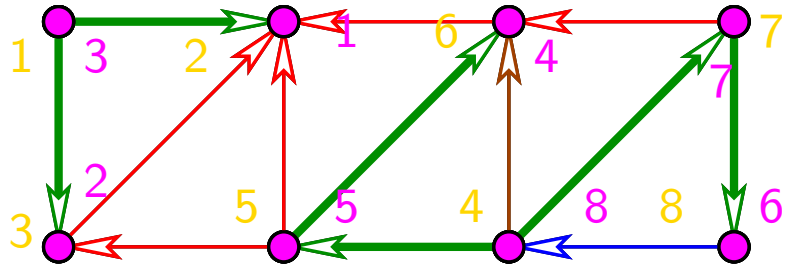
(3) Ha uv visszaél, akkor $m(u) > m(v)$ és $b(u) < b(v)$.

(4) Ha uv keresztél, akkor $m(u) > m(v)$ és $b(u) > b(v)$.

Biz: $m(u) < m(v)$ esetén a DFS miatt v az u leszármazottja lenne. Ezért $m(u) > m(v)$. Ha u-t a v befejezése előtt érnénk el, akkor u a v leszármazottja lenne. Ezért az alábbi sorrendben történik u és v evolúciója: v elérése, v befejezése, u elérése, u befejezése.

Depth First Search





Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

Megf: Tfh a G gráf éleit DFS után osztályoztuk.

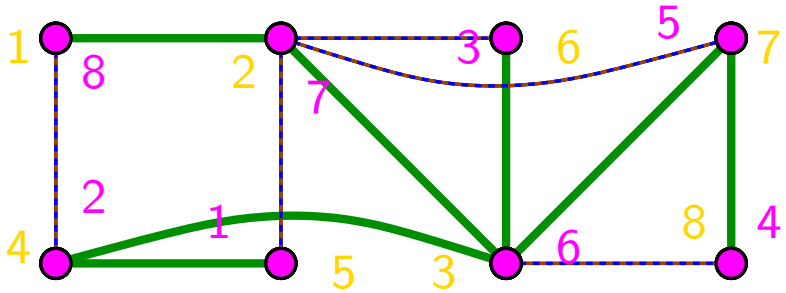
(1) Ha uv faél, akkor $m(u) < m(v)$ és $b(u) > b(v)$.

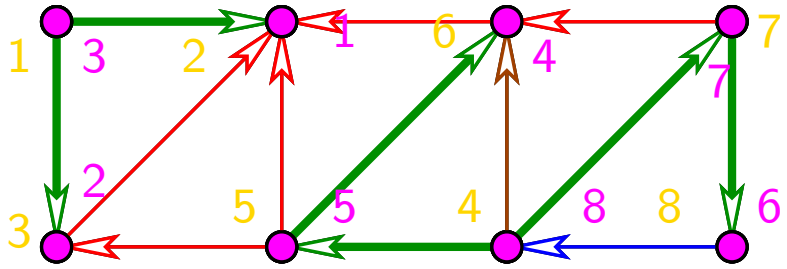
(2) Ha uv előreél, akkor $m(u) < m(v)$ és $b(u) > b(v)$.

(3) Ha uv visszaél, akkor $m(u) > m(v)$ és $b(u) < b(v)$.

(4) Ha uv keresztél, akkor $m(u) > m(v)$ és $b(u) > b(v)$.

Depth First Search





Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

Megf: Tfh a G gráf éleit DFS után osztályoztuk.

(1) Ha uv faél, akkor $m(u) < m(v)$ és $b(u) > b(v)$.

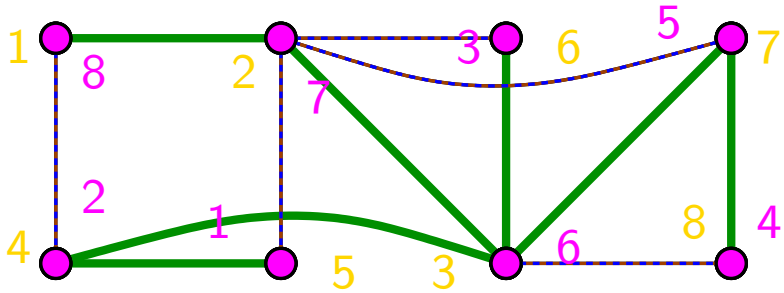
(2) Ha uv előreél, akkor $m(u) < m(v)$ és $b(u) > b(v)$.

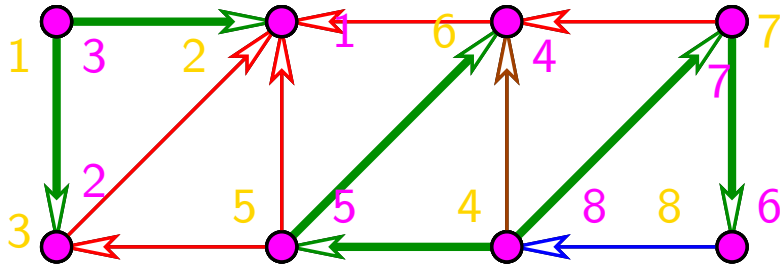
(3) Ha uv visszaél, akkor $m(u) > m(v)$ és $b(u) < b(v)$.

(4) Ha uv keresztél, akkor $m(u) > m(v)$ és $b(u) > b(v)$.

(5) **Irányítatlan gráf DFS bejárása után nincs keresztél.**

Depth First Search





Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

Megf: Tfh a G gráf éleit DFS után osztályoztuk.

(1) Ha uv faél, akkor $m(u) < m(v)$ és $b(u) > b(v)$.

(2) Ha uv előreél, akkor $m(u) < m(v)$ és $b(u) > b(v)$.

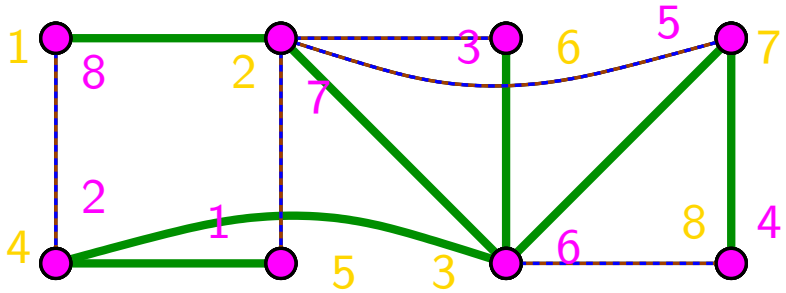
(3) Ha uv visszaél, akkor $m(u) > m(v)$ és $b(u) < b(v)$.

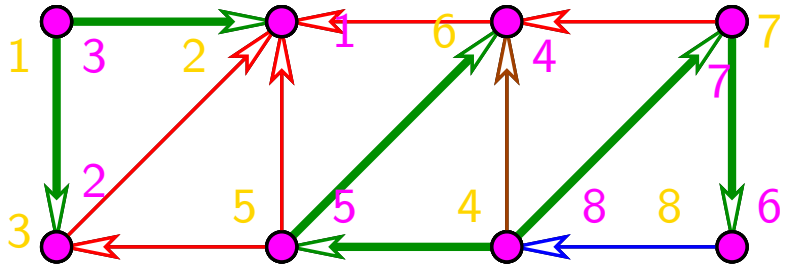
(4) Ha uv keresztél, akkor $m(u) > m(v)$ és $b(u) > b(v)$.

(5) **Irányítatlan gráf DFS bejárása után nincs keresztél.**

Biz: Indirekt. Ha uv keresztél, akkor (4) miatt $m(u) > m(v)$, továbbá vu is keresztél, ezért $m(v) > m(u)$. Ellentmondás. □

Depth First Search





Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

Megf: Tfh a G gráf éleit DFS után osztályoztuk.

(1) Ha uv faél, akkor $m(u) < m(v)$ és $b(u) > b(v)$.

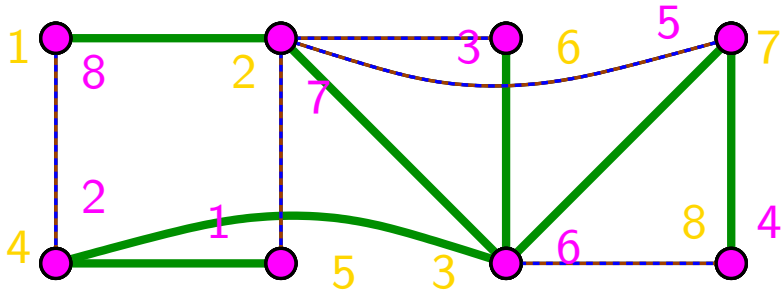
(2) Ha uv előreél, akkor $m(u) < m(v)$ és $b(u) > b(v)$.

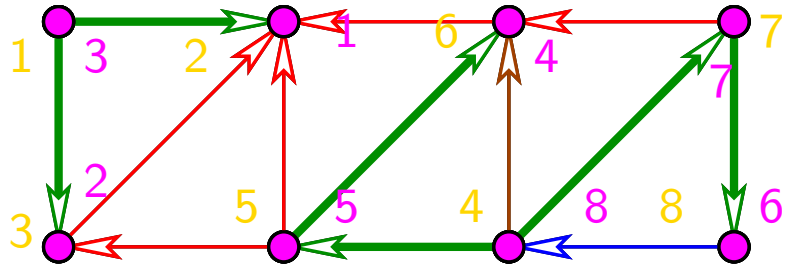
(3) Ha uv visszaél, akkor $m(u) > m(v)$ és $b(u) < b(v)$.

(4) Ha uv keresztél, akkor $m(u) > m(v)$ és $b(u) > b(v)$.

(5) **Irányítatlan gráf DFS bejárása után nincs keresztél.**

Depth First Search





Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

Megf: Tfh a G gráf éleit DFS után osztályoztuk.

(1) Ha uv faél, akkor $m(u) < m(v)$ és $b(u) > b(v)$.

(2) Ha uv előreél, akkor $m(u) < m(v)$ és $b(u) > b(v)$.

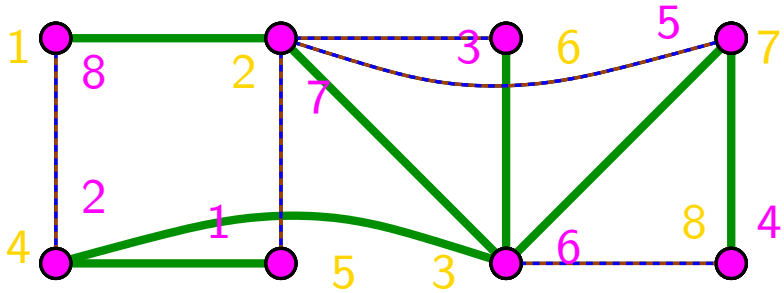
(3) Ha uv visszaél, akkor $m(u) > m(v)$ és $b(u) < b(v)$.

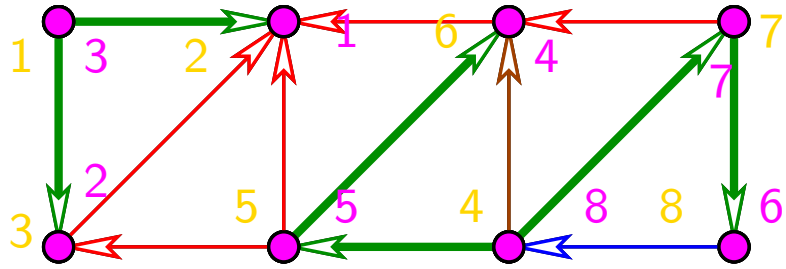
(4) Ha uv keresztél, akkor $m(u) > m(v)$ és $b(u) > b(v)$.

(5) **Irányítatlan gráf DFS bejárása után nincs keresztél.**

(6) **Ha DFS után van visszaél, $\Rightarrow$ G-ben van irányított kör.**

Depth First Search





Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

Megf: Tfh a G gráf éleit DFS után osztályoztuk.

(1) Ha uv faél, akkor $m(u) < m(v)$ és $b(u) > b(v)$.

(2) Ha uv előreél, akkor $m(u) < m(v)$ és $b(u) > b(v)$.

(3) Ha uv visszaél, akkor $m(u) > m(v)$ és $b(u) < b(v)$.

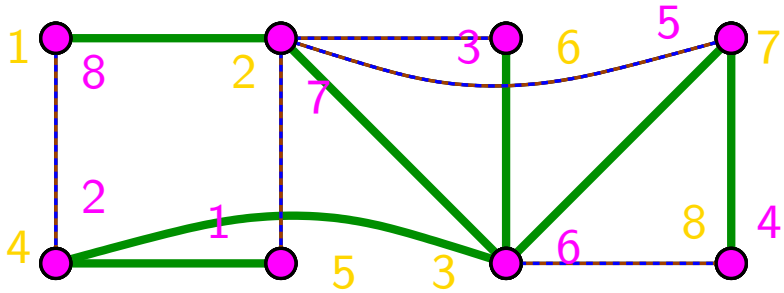
(4) Ha uv keresztél, akkor $m(u) > m(v)$ és $b(u) > b(v)$.

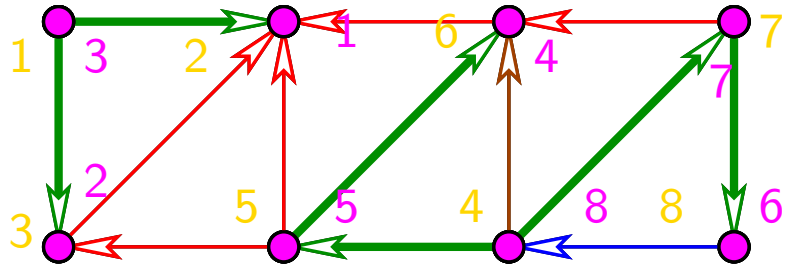
(5) **Irányítatlan gráf DFS bejárása után nincs keresztél.**

(6) **Ha DFS után van visszaél, $\Rightarrow$ G-ben van irányított kör.**

Biz: Irányított gráf bármely bejárása esetén a bejárási fa visszaélhez tartozó alapköre irányított kör az eredeti gráfban. □

Depth First Search





Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

Megf: Tfh a G gráf éleit DFS után osztályoztuk.

(1) Ha uv faél, akkor $m(u) < m(v)$ és $b(u) > b(v)$.

(2) Ha uv előreél, akkor $m(u) < m(v)$ és $b(u) > b(v)$.

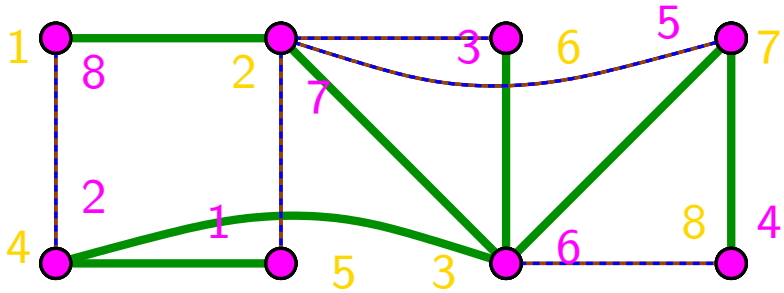
(3) Ha uv visszaél, akkor $m(u) > m(v)$ és $b(u) < b(v)$.

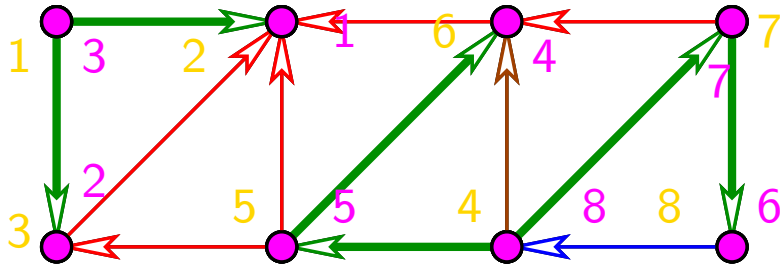
(4) Ha uv keresztél, akkor $m(u) > m(v)$ és $b(u) > b(v)$.

(5) **Irányítatlan gráf DFS bejárása után nincs keresztél.**

(6) **Ha DFS után van visszaél, $\Rightarrow$ G-ben van irányított kör.**

Depth First Search





Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

Megf: Tfh a G gráf éleit DFS után osztályoztuk.

(1) Ha uv faél, akkor $m(u) < m(v)$ és $b(u) > b(v)$.

(2) Ha uv előreél, akkor $m(u) < m(v)$ és $b(u) > b(v)$.

(3) Ha uv visszaél, akkor $m(u) > m(v)$ és $b(u) < b(v)$.

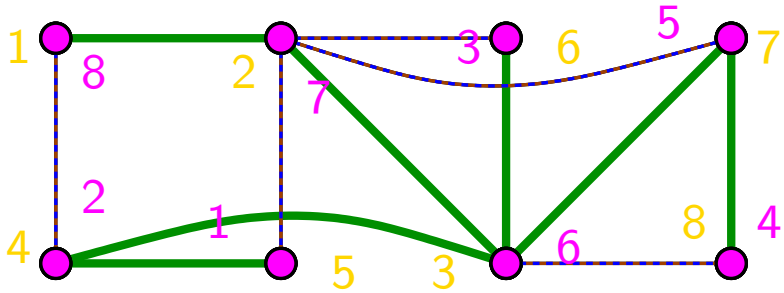
(4) Ha uv keresztél, akkor $m(u) > m(v)$ és $b(u) > b(v)$.

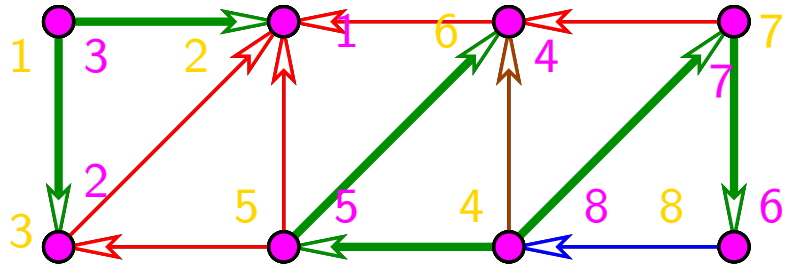
(5) **Irányítatlan gráf DFS bejárása után nincs keresztél.**

(6) **Ha DFS után van visszaél, $\Rightarrow$ G-ben van irányított kör.**

(7) **Ha DFS után nincs visszaél $\Rightarrow$ G-ben nincs irányított kör.**

Depth First Search





Mélységi bejárás (DFS)

Mindig a legutolsónak elért csúcsból folytatjuk a bejárást.

Mélységi és befejezési számozás: DFS után $m(v)$ ill. $b(v)$ a v csúcs elérési ill. befejezési sorrendben kapott sorszáma.

Megf: Tfh a G gráf éleit DFS után osztályoztuk.

(1) Ha uv faél, akkor $m(u) < m(v)$ és $b(u) > b(v)$.

(2) Ha uv előreél, akkor $m(u) < m(v)$ és $b(u) > b(v)$.

(3) Ha uv visszaél, akkor $m(u) > m(v)$ és $b(u) < b(v)$.

(4) Ha uv keresztél, akkor $m(u) > m(v)$ és $b(u) > b(v)$.

(5) **Irányítatlan gráf DFS bejárása után nincs keresztél.**

(6) **Ha DFS után van visszaél, $\Rightarrow$ G-ben van irányított kör.**

(7) **Ha DFS után nincs visszaél $\Rightarrow$ G-ben nincs irányított kör.**

Biz: Minden irányított körben van olyan uv él, amire $b(u) < b(v)$. Ez az él csakis visszaél lehet. Így ha nincs visszaél, ir. kör sincs. □

Directed Acyclic Graphs

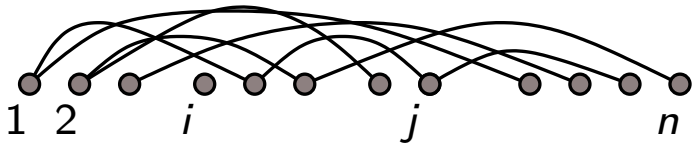
Def: A $G = (V, E)$ irányított gráf aciklikus (más néven DAG), ha G nem tartalmaz irányított kört.

Directed Acyclic Graphs

Def: A $G = (V, E)$ irányított gráf aciklikus (más néven DAG), ha G nem tartalmaz irányított kört.

Példa: DAG-ot pl úgy kaphatunk, hogy egy G irányítatlan gráf csúcsait csupa különböző számmal megszámozzuk, és minden élt a kisebb számot viselő csúcsból a nagyobba irányítunk.

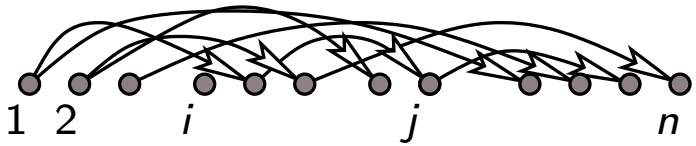
Directed Acyclic Graphs



Def: A $G = (V, E)$ irányított gráf aciklikus (más néven DAG), ha G nem tartalmaz irányított kört.

Példa: DAG-ot pl úgy kaphatunk, hogy egy G irányítatlan gráf csúcsait csupa különböző számmal megszámozzuk, és minden élt a kisebb számot viselő csúcsból a nagyobba irányítunk.

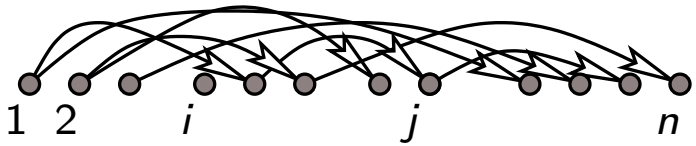
Directed Acyclic Graphs



Def: A $G = (V, E)$ irányított gráf aciklikus (más néven DAG), ha G nem tartalmaz irányított kört.

Példa: DAG-ot pl úgy kaphatunk, hogy egy G irányítatlan gráf csúcsait csupa különböző számmal megszámozzuk, és minden élt a kisebb számot viselő csúcsból a nagyobba irányítunk.

Directed Acyclic Graphs



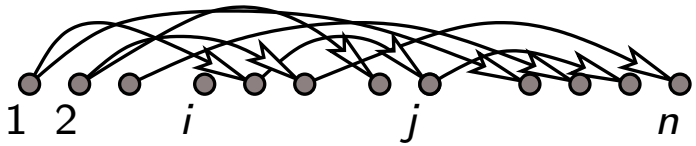
Def: A $G = (V, E)$ irányított gráf aciklikus (más néven DAG), ha G nem tartalmaz irányított kört.

Példa: DAG-ot pl úgy kaphatunk, hogy egy G irányítatlan gráf csúcsait csupa különböző számmal megszámozzuk, és minden élt a kisebb számot viselő csúcsból a nagyobba irányítunk.

Ha ugyanis lenne az így megirányított gráfban irányított kör, akkor az élei mentén a számozás végig növekedne, ami lehetetlen.

Azt fogjuk igazolni, hogy a fenti példa minden DAG-ot leír.

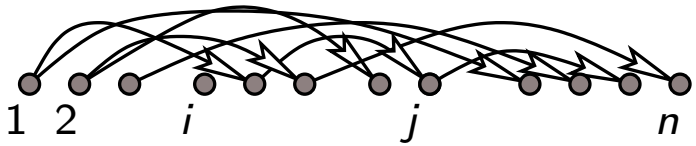
Directed Acyclic Graphs



Def: A $G = (V, E)$ irányított gráf aciklikus (más néven DAG), ha G nem tartalmaz irányított kört.

Példa: DAG-ot pl úgy kaphatunk, hogy egy G irányítatlan gráf csúcsait csupa különböző számmal megszámozzuk, és minden élt a kisebb számot viselő csúcsból a nagyobba irányítunk.

Directed Acyclic Graphs

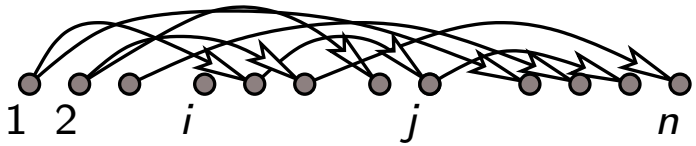


Def: A $G = (V, E)$ irányított gráf aciklikus (más néven DAG), ha G nem tartalmaz irányított kört.

Példa: DAG-ot pl úgy kaphatunk, hogy egy G irányítatlan gráf csúcsait csupa különböző számmal megszámozzuk, és minden élt a kisebb számot viselő csúcsból a nagyobba irányítunk.

Def: A $G = (V, E)$ irányított gráf csúcsainak topologikus sorrendje alatt a csúcsok olyan sorrendjét értjük, amire igaz, hogy minden irányított él a sorban előbb álló csúcsból vezet a sorban későbbi csúcsba. (Azaz $V = \{v_1, v_2, \ldots, v_n\}$, ahol $v_i v_j \in E \Rightarrow i < j$.)

Directed Acyclic Graphs



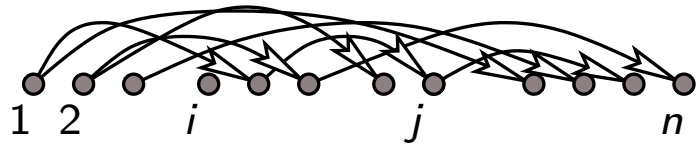
Def: A $G = (V, E)$ irányított gráf aciklikus (más néven DAG), ha G nem tartalmaz irányított kört.

Példa: DAG-ot pl úgy kaphatunk, hogy egy G irányítatlan gráf csúcsait csupa különböző számmal megszámozzuk, és minden élt a kisebb számot viselő csúcsból a nagyobba irányítunk.

Def: A $G = (V, E)$ irányított gráf csúcsainak topologikus sorrendje alatt a csúcsok olyan sorrendjét értjük, amire igaz, hogy minden irányított él a sorban előbb álló csúcsból vezet a sorban későbbi csúcsba. (Azaz $V = \{v_1, v_2, \ldots, v_n\}$, ahol $v_i v_j \in E \Rightarrow i < j$.)

Tétel: (G irányított gráf DAG) $\Longleftrightarrow$ ($V(G)$-nek $\exists$ top. sorrendje).

Directed Acyclic Graphs



Def: A $G = (V, E)$ irányított gráf aciklikus (más néven DAG), ha G nem tartalmaz irányított kört.

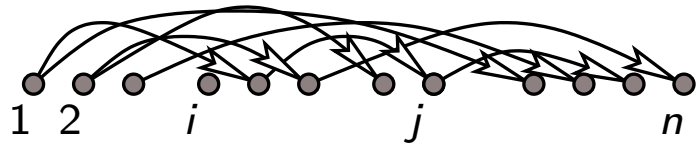
Példa: DAG-ot pl úgy kaphatunk, hogy egy G irányítatlan gráf csúcsait csupa különböző számmal megszámozzuk, és minden élt a kisebb számot viselő csúcsból a nagyobba irányítunk.

Def: A $G = (V, E)$ irányított gráf csúcsainak topologikus sorrendje alatt a csúcsok olyan sorrendjét értjük, amire igaz, hogy minden irányított él a sorban előbb álló csúcsból vezet a sorban későbbi csúcsba. (Azaz $V = \{v_1, v_2, \ldots, v_n\}$, ahol $v_i v_j \in E \Rightarrow i < j$.)

Tétel: (G irányított gráf DAG) $\Longleftrightarrow$ ($V(G)$-nek $\exists$ top. sorrendje).

Biz: $\Leftarrow$: Tfh $\exists$ top. sorrend. Láttuk, hogy G ekkor DAG. ✓

Directed Acyclic Graphs



Def: A $G = (V, E)$ irányított gráf aciklikus (más néven DAG), ha G nem tartalmaz irányított kört.

Példa: DAG-ot pl úgy kaphatunk, hogy egy G irányítatlan gráf csúcsait csupa különböző számmal megszámozzuk, és minden élt a kisebb számot viselő csúcsból a nagyobba irányítunk.

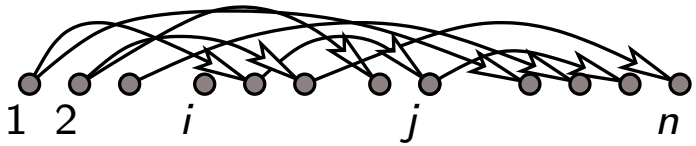
Def: A $G = (V, E)$ irányított gráf csúcsainak topologikus sorrendje alatt a csúcsok olyan sorrendjét értjük, amire igaz, hogy minden irányított él a sorban előbb álló csúcsból vezet a sorban későbbi csúcsba. (Azaz $V = \{v_1, v_2, \ldots, v_n\}$, ahol $v_i v_j \in E \Rightarrow i < j$.)

Tétel: (G irányított gráf DAG) $\Longleftrightarrow$ ($V(G)$-nek $\exists$ top. sorrendje).

Biz: $\Leftarrow$: Tfh $\exists$ top. sorrend. Láttuk, hogy G ekkor DAG. ✓

$\Rightarrow$: Most tfh G DAG, és futtassunk rajta egy DFS-t. Láttuk, hogy a DFS után nem lesz visszaél, tehát $b(u) > b(v)$ teljesül G minden uv élére. Ezért a csúcsok befejezési sorrendjének megfordítása a G csúcsainak egy topologikus sorrendje. □

Directed Acyclic Graphs



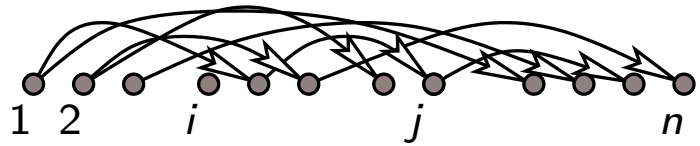
Def: A $G = (V, E)$ irányított gráf aciklikus (más néven DAG), ha G nem tartalmaz irányított kört.

Példa: DAG-ot pl úgy kaphatunk, hogy egy G irányítatlan gráf csúcsait csupa különböző számmal megszámozzuk, és minden élt a kisebb számot viselő csúcsból a nagyobba irányítunk.

Def: A $G = (V, E)$ irányított gráf csúcsainak topologikus sorrendje alatt a csúcsok olyan sorrendjét értjük, amire igaz, hogy minden irányított él a sorban előbb álló csúcsból vezet a sorban későbbi csúcsba. (Azaz $V = \{v_1, v_2, \ldots, v_n\}$, ahol $v_i v_j \in E \Rightarrow i < j$.)

Tétel: (G irányított gráf DAG) $\Longleftrightarrow$ ($V(G)$-nek $\exists$ top. sorrendje).

Directed Acyclic Graphs



Def: A $G = (V, E)$ irányított gráf aciklikus (más néven DAG), ha G nem tartalmaz irányított kört.

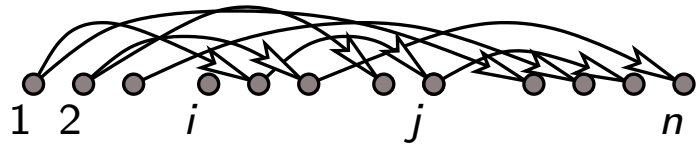
Példa: DAG-ot pl úgy kaphatunk, hogy egy G irányítatlan gráf csúcsait csupa különböző számmal megszámozzuk, és minden élt a kisebb számot viselő csúcsból a nagyobba irányítunk.

Def: A $G = (V, E)$ irányított gráf csúcsainak topologikus sorrendje alatt a csúcsok olyan sorrendjét értjük, amire igaz, hogy minden irányított él a sorban előbb álló csúcsból vezet a sorban későbbi csúcsba. (Azaz $V = \{v_1, v_2, \ldots, v_n\}$, ahol $v_i v_j \in E \Rightarrow i < j$.)

Tétel: (G irányított gráf DAG) $\Longleftrightarrow$ ($V(G)$-nek $\exists$ top. sorrendje).

Köv: Irányított gráf aciklikussága DFS-sel gyorsan eldönthető: ha van visszaél, akkor a visszaél DFS-fabeli alapköre G egy irányított köre. Ez bizonyítja, hogy G nem DAG. Ha pedig nincs visszaél a DFS után, akkor a fordított befejezési sorrend a G egy topologikus sorrendje. Ez világosan mutatja, hogy G DAG.

Directed Acyclic Graphs



Def: A $G = (V, E)$ irányított gráf aciklikus (más néven DAG), ha G nem tartalmaz irányított kört.

Példa: DAG-ot pl úgy kaphatunk, hogy egy G irányítatlan gráf csúcsait csupa különböző számmal megszámozzuk, és minden élt a kisebb számot viselő csúcsból a nagyobba irányítunk.

Def: A $G = (V, E)$ irányított gráf csúcsainak topologikus sorrendje alatt a csúcsok olyan sorrendjét értjük, amire igaz, hogy minden irányított él a sorban előbb álló csúcsból vezet a sorban későbbi csúcsba. (Azaz $V = \{v_1, v_2, \ldots, v_n\}$, ahol $v_i v_j \in E \Rightarrow i < j$.)

Tétel: (G irányított gráf DAG) $\Longleftrightarrow$ ($V(G)$-nek $\exists$ top. sorrendje).

Köv: Irányított gráf aciklikussága DFS-sel gyorsan eldönthető: ha van visszaél, akkor a visszaél DFS-fabeli alapköre G egy irányított köre. Ez bizonyítja, hogy G nem DAG. Ha pedig nincs visszaél a DFS után, akkor a fordított befejezési sorrend a G egy topologikus sorrendje. Ez világosan mutatja, hogy G DAG.

Megj: DAG egy topologikus sorrendjét forráskeresések és forrástörlések alkalmazásával is megtalálhatjuk.

Leghosszabb út keresése

Első pillantásra haszontalannak tűnik, de matematikailag érdekes kérdés egy gráf két csúcsa között a leghosszabb út megtalálása.

Leghosszabb út keresése

Első pillantásra haszontalannak tűnik, de matematikailag érdekes kérdés egy gráf két csúcsa között a leghosszabb út megtalálása. Legrövidebb utat tudunk keresni; tudunk vajon leghosszabbat is?

Leghosszabb út keresése

Első pillantásra haszontalannak tűnik, de matematikailag érdekes kérdés egy gráf két csúcsa között a leghosszabb út megtalálása. Legrövidebb utat tudunk keresni; tudunk vajon leghosszabbat is?

Ötlet: Az $\ell'(uv) = -\ell(uv)$ élhosszokkal a leghosszabb utak legrövidebbekké válnak. Olyanokat pedig tudunk keresni.

Leghosszabb út keresése

Első pillantásra haszontalannak tűnik, de matematikailag érdekes kérdés egy gráf két csúcsa között a leghosszabb út megtalálása. Legrövidebb utat tudunk keresni; tudunk vajon leghosszabbat is?

Ötlet: Az $\ell'(uv) = -\ell(uv)$ élhosszokkal a leghosszabb utak legrövidebbekké válnak. Olyanokat pedig tudunk keresni.

Gond: Az ismert módszerek csak konzervatív élhosszokra működnek. Ha azonban van a gráfban irányított kör, akkor ez az ötlet általában nem segít. Irányított esetben ugyanis kizárólag akkor konzervatív egy csupa negatív élhosszokat tartalmazó hosszfüggvény, ha a gráfban nincs irányított kör, azaz ha G DAG. Ekkor a Ford vagy Floyd algoritmusok bármelyike használható.

Leghosszabb út keresése

Első pillantásra haszontalannak tűnik, de matematikailag érdekes kérdés egy gráf két csúcsa között a leghosszabb út megtalálása. Legrövidebb utat tudunk keresni; tudunk vajon leghosszabbat is?

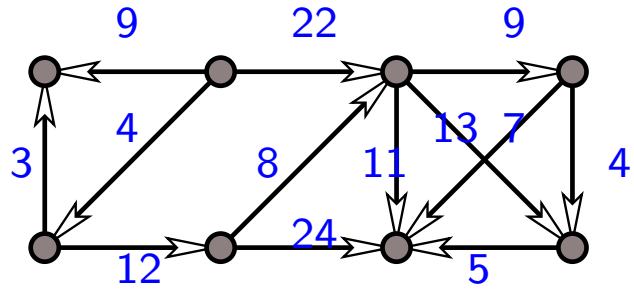
Ötlet: Az $\ell'(uv) = -\ell(uv)$ élhosszokkal a leghosszabb utak legrövidebbekké válnak. Olyanokat pedig tudunk keresni.

Gond: Az ismert módszerek csak konzervatív élhosszokra működnek. Ha azonban van a gráfban irányított kör, akkor ez az ötlet általában nem segít. Irányított esetben ugyanis kizárólag akkor konzervatív egy csupa negatív élhosszokat tartalmazó hosszfüggvény, ha a gráfban nincs irányított kör, azaz ha G DAG. Ekkor a Ford vagy Floyd algoritmusok bármelyike használható.

Jó hír: Van egy még gyorsabb módszer: a dinamikus programozás. Ennek segítségével tetsz. G DAG minden v csúcsához ki tudjuk számítani a v-be vezető leghosszabb utat. (Sőt! ...)

Leghosszabb út keresése

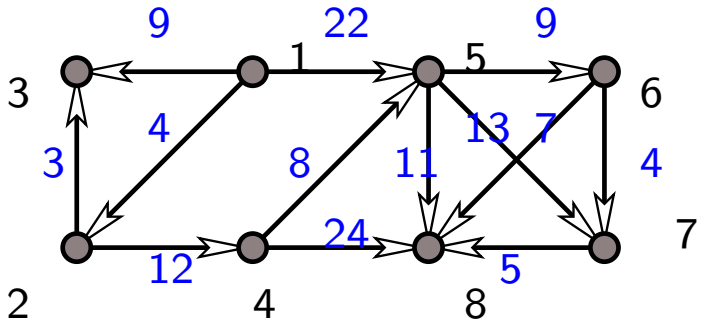
Leghosszabb út keresése



Leghosszabb út DAG-ban Input: $G = (V, E)$ DAG, $\ell : E \to \mathbb{R}$.
Output: $\max\{\ell(P) : P$ v-be vezető út$\}$ minden $v \in V$ csúcsra.

Működés: $\boxed{1}$ $V = \{v_1, v_2, \ldots, v_n\}$ top. sorrend meghatározása.
$\boxed{2}$ $i = 1, 2, \ldots, n$: $f(v_i) = \max\{\max\{f(v_j) + \ell(v_j v_i) : v_j v_i \in E\}, 0\}$
Output: $f(v)$ $\forall v \in V$

Leghosszabb út keresése



Leghosszabb út DAG-ban Input: $G = (V, E)$ DAG, $\ell : E \to \mathbb{R}$.

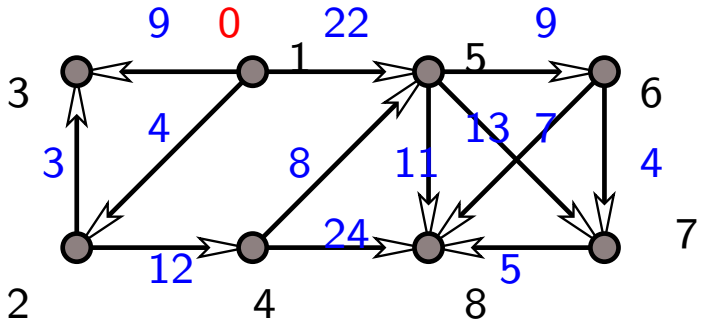
Output: $\max\{\ell(P) : P$ v-be vezető út$\}$ minden $v \in V$ csúcsra.

Működés: $\boxed{1}$ $V = \{v_1, v_2, \ldots, v_n\}$ top. sorrend meghatározása.

$\boxed{2}$ $i = 1, 2, \ldots, n$: $f(v_i) = \max\{\max\{f(v_j) + \ell(v_j v_i) : v_j v_i \in E\}, 0\}$

Output: $f(v)$ $\forall v \in V$

Leghosszabb út keresése



Leghosszabb út DAG-ban Input: $G = (V, E)$ DAG, $\ell : E \to \mathbb{R}$.

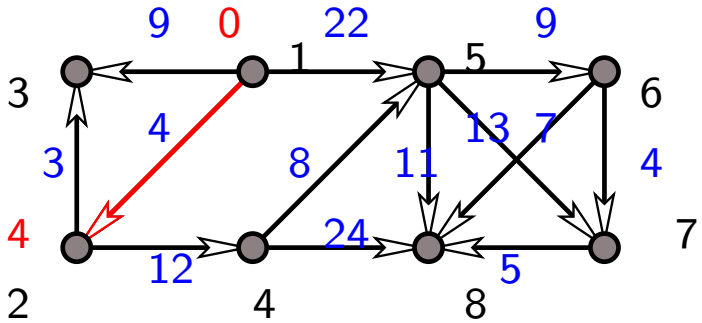
Output: $\max\{\ell(P) : P$ v-be vezető út$\}$ minden $v \in V$ csúcsra.

Működés: $\boxed{1}$ $V = \{v_1, v_2, \ldots, v_n\}$ top. sorrend meghatározása.

$\boxed{2}$ $i = 1, 2, \ldots, n$: $f(v_i) = \max\{\max\{f(v_j) + \ell(v_j v_i) : v_j v_i \in E\}, 0\}$

Output: $f(v)$ $\forall v \in V$

Leghosszabb út keresése



Leghosszabb út DAG-ban Input: $G = (V, E)$ DAG, $\ell : E \to \mathbb{R}$.

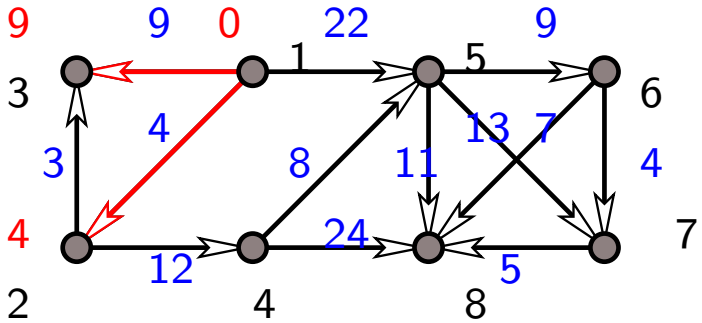
Output: $\max\{\ell(P) : P$ v-be vezető út$\}$ minden $v \in V$ csúcsra.

Működés: $\boxed{1}$ $V = \{v_1, v_2, \ldots, v_n\}$ top. sorrend meghatározása.

$\boxed{2}$ $i = 1, 2, \ldots, n$: $f(v_i) = \max\{\max\{f(v_j) + \ell(v_j v_i) : v_j v_i \in E\}, 0\}$

Output: $f(v)$ $\forall v \in V$

Leghosszabb út keresése



Leghosszabb út DAG-ban Input: $G = (V, E)$ DAG, $\ell : E \to \mathbb{R}$.

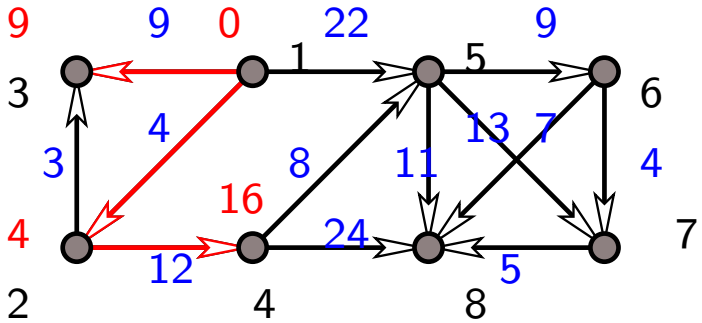
Output: $\max\{\ell(P) : P$ v-be vezető út$\}$ minden $v \in V$ csúcsra.

Működés: $\boxed{1}$ $V = \{v_1, v_2, \ldots, v_n\}$ top. sorrend meghatározása.

$\boxed{2}$ $i = 1, 2, \ldots, n$: $f(v_i) = \max\{\max\{f(v_j) + \ell(v_j v_i) : v_j v_i \in E\}, 0\}$

Output: $f(v)$ $\forall v \in V$

Leghosszabb út keresése



Leghosszabb út DAG-ban Input: $G = (V, E)$ DAG, $\ell : E \to \mathbb{R}$.

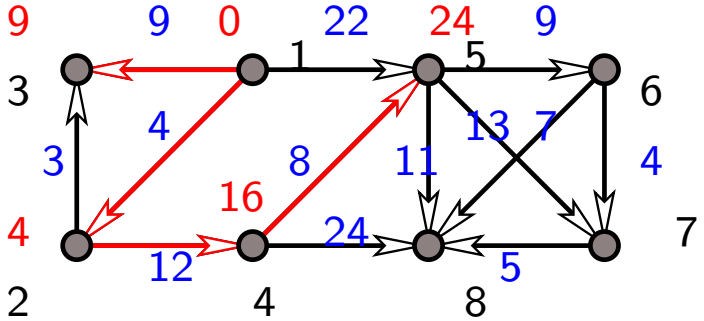
Output: $\max\{\ell(P) : P$ v-be vezető út$\}$ minden $v \in V$ csúcsra.

Működés: $\boxed{1}$ $V = \{v_1, v_2, \ldots, v_n\}$ top. sorrend meghatározása.

$\boxed{2}$ $i = 1, 2, \ldots, n$: $f(v_i) = \max\{\max\{f(v_j) + \ell(v_j v_i) : v_j v_i \in E\}, 0\}$

Output: $f(v)$ $\forall v \in V$

Leghosszabb út keresése



Leghosszabb út DAG-ban Input: $G = (V, E)$ DAG, $\ell : E \to \mathbb{R}$.

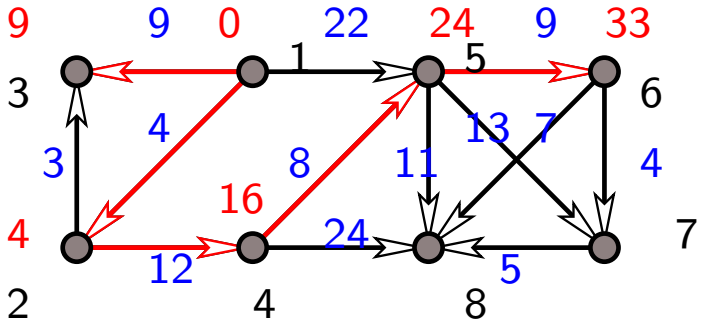
Output: $\max\{\ell(P) : P$ v-be vezető út$\}$ minden $v \in V$ csúcsra.

Működés: $\boxed{1}$ $V = \{v_1, v_2, \ldots, v_n\}$ top. sorrend meghatározása.

$\boxed{2}$ $i = 1, 2, \ldots, n$: $f(v_i) = \max\{\max\{f(v_j) + \ell(v_j v_i) : v_j v_i \in E\}, 0\}$

Output: $f(v)$ $\forall v \in V$

Leghosszabb út keresése



Leghosszabb út DAG-ban Input: $G = (V, E)$ DAG, $\ell : E \to \mathbb{R}$.

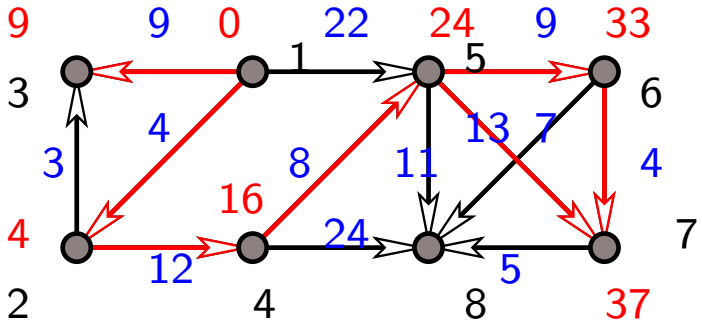
Output: $\max\{\ell(P) : P$ v-be vezető út$\}$ minden $v \in V$ csúcsra.

Működés: $\boxed{1}$ $V = \{v_1, v_2, \ldots, v_n\}$ top. sorrend meghatározása.

$\boxed{2}$ $i = 1, 2, \ldots, n$: $f(v_i) = \max\{\max\{f(v_j) + \ell(v_j v_i) : v_j v_i \in E\}, 0\}$

Output: $f(v)$ $\forall v \in V$

Leghosszabb út keresése



Leghosszabb út DAG-ban Input: $G = (V, E)$ DAG, $\ell : E \to \mathbb{R}$.

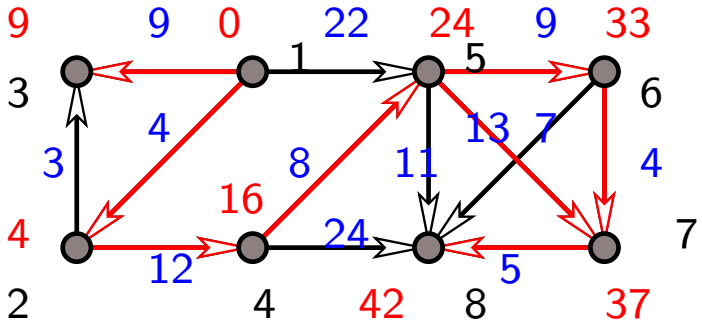
Output: $\max\{\ell(P) : P$ v-be vezető út$\}$ minden $v \in V$ csúcsra.

Működés: $\boxed{1}$ $V = \{v_1, v_2, \ldots, v_n\}$ top. sorrend meghatározása.

$\boxed{2}$ $i = 1, 2, \ldots, n$: $f(v_i) = \max\{\max\{f(v_j) + \ell(v_j v_i) : v_j v_i \in E\}, 0\}$

Output: $f(v)$ $\forall v \in V$

Leghosszabb út keresése



Leghosszabb út DAG-ban Input: $G = (V, E)$ DAG, $\ell : E \to \mathbb{R}$.

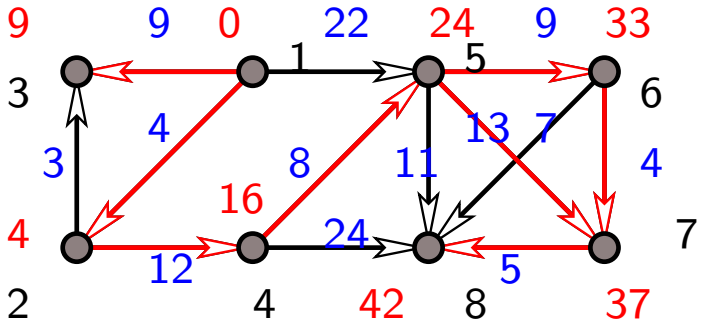
Output: $\max\{\ell(P) : P$ v-be vezető út$\}$ minden $v \in V$ csúcsra.

Működés: $\boxed{1}$ $V = \{v_1, v_2, \ldots, v_n\}$ top. sorrend meghatározása.

$\boxed{2}$ $i = 1, 2, \ldots, n$: $f(v_i) = \max\{\max\{f(v_j) + \ell(v_j v_i) : v_j v_i \in E\}, 0\}$

Output: $f(v)$ $\forall v \in V$

Leghosszabb út keresése



Leghosszabb út DAG-ban Input: $G = (V, E)$ DAG, $\ell : E \to \mathbb{R}$.

Output: $\max\{\ell(P) : P$ v-be vezető út$\}$ minden $v \in V$ csúcsra.

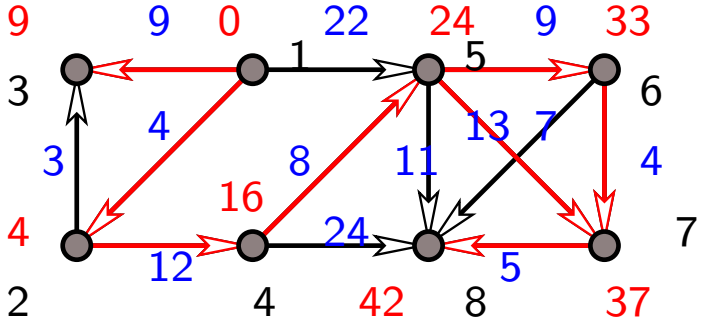
Működés: $\boxed{1}$ $V = \{v_1, v_2, \ldots, v_n\}$ top. sorrend meghatározása.

$\boxed{2}$ $i = 1, 2, \ldots, n$: $f(v_i) = \max\{\max\{f(v_j) + \ell(v_j v_i) : v_j v_i \in E\}, 0\}$

Output: $f(v)$ $\forall v \in V$

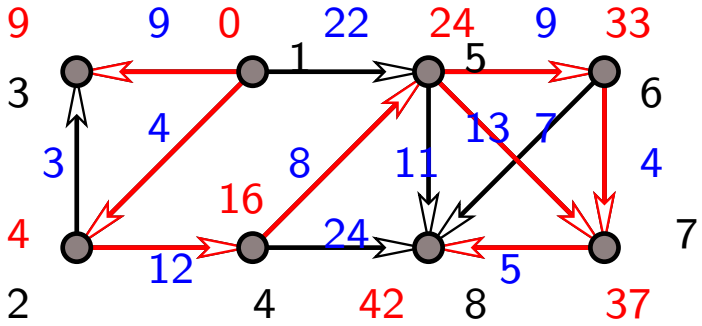
Helyesség: Ha a v_i-be vezető leghosszabb út utolsó előtti csúcsa v_j, akkor $f(v_i) = f(v_j) + \ell(v_j v_i)$. □

Leghosszabb út keresése



Leghosszabb út DAG-ban Input: $G = (V, E)$ DAG, $\ell : E \to \mathbb{R}$.
Output: $\max\{\ell(P) : P$ v-be vezető út$\}$ minden $v \in V$ csúcsra.
Működés: $\boxed{1}$ $V = \{v_1, v_2, \ldots, v_n\}$ top. sorrend meghatározása.
$\boxed{2}$ $i = 1, 2, \ldots, n$: $f(v_i) = \max\{\max\{f(v_j) + \ell(v_j v_i) : v_j v_i \in E\}, 0\}$
Output: $f(v)$ $\forall v \in V$
Helyesség: Ha a v_i-be vezető leghosszabb út utolsó előtti csúcsa v_j, akkor $f(v_i) = f(v_j) + \ell(v_j v_i)$. □
Megj: Ha a fenti algoritmusban minden csúcsra megjelöljük az $f(v)$ értéket beállító élt (éleket), akkor a megjelölt élek minden v csúcsba megadnak egy leghosszabb utat. Sőt: a v-be vezető leghosszabb utak mindegyike megkapható így.

Leghosszabb út keresése



Leghosszabb út DAG-ban Input: $G = (V, E)$ DAG, $\ell : E \to \mathbb{R}$.
Output: $\max\{\ell(P) : P$ v-be vezető út$\}$ minden $v \in V$ csúcsra.
Működés: $\boxed{1}$ $V = \{v_1, v_2, \ldots, v_n\}$ top. sorrend meghatározása.
$\boxed{2}$ $i = 1, 2, \ldots, n$: $f(v_i) = \max\{\max\{f(v_j) + \ell(v_j v_i) : v_j v_i \in E\}, 0\}$
Output: $f(v)$ $\forall v \in V$
Helyesség: Ha a v_i-be vezető leghosszabb út utolsó előtti csúcsa v_j, akkor $f(v_i) = f(v_j) + \ell(v_j v_i)$. □
Megj: Ha a fenti algoritmusban minden csúcsra megjelöljük az $f(v)$ értéket beállító élt (éleket), akkor a megjelölt élek minden v csúcsba megadnak egy leghosszabb utat. Sőt: a v-be vezető leghosszabb utak mindegyike megkapható így.
Kínzó kérdés: Van-e bármi értelme leghosszabb utakat keresni?

A PERT probléma

Egy $a, b, \ldots$ tevékenységekből álló projektet kell végrehajtanunk.

Precedenciafeltételek: bizonyos (u, v) párok esetén előírás, hogy az u tevékenységet a v előtt kell elvégezni, ezért v az u kezdetét követően $c(uv)$ időkorlát elteltével kezdhető.

Cél: minden v tevékenységehez olyan $k(v) \geq 0$ kezdési időpont meghatározása, ami nem sérti a preferenciafeltételeket, és a projekt végrehajtási ideje (vagyis a legnagyobb $k(v)$ érték) minimális.

A PERT probléma

Egy $a, b, \ldots$ tevékenységekből álló projektet kell végrehajtanunk.
Precedenciafeltételek: bizonyos (u, v) párok esetén előírás, hogy az u tevékenységet a v előtt kell elvégezni, ezért v az u kezdetét követően $c(uv)$ időkorlát elteltével kezdhető.
Cél: minden v tevékenységehez olyan $k(v) \geq 0$ kezdési időpont meghatározása, ami nem sérti a preferenciafeltételeket, és a projekt végrehajtási ideje (vagyis a legnagyobb $k(v)$ érték) minimális.
G **irányított gráf** csúcsai a tevékenységek, élei pedig a precedenciafeltételek, az uv él hossza $c(uv)$.
Megf: (1) Ha G nem DAG, akkor a projekt nem hajtható végre.

A PERT probléma

Egy $a, b, \ldots$ tevékenységekből álló projektet kell végrehajtanunk.
Precedenciafeltételek: bizonyos (u, v) párok esetén előírás, hogy az u tevékenységet a v előtt kell elvégezni, ezért v az u kezdetét követően $c(uv)$ időkorlát elteltével kezdhető.
Cél: minden v tevékenységehez olyan $k(v) \geq 0$ kezdési időpont meghatározása, ami nem sérti a preferenciafeltételeket, és a projekt végrehajtási ideje (vagyis a legnagyobb $k(v)$ érték) minimális.
G **irányított gráf** csúcsai a tevékenységek, élei pedig a precedenciafeltételek, az uv él hossza $c(uv)$.
Megf: (1) Ha G nem DAG, akkor a projekt nem hajtható végre.
(2) Ha G DAG, akkor minden v tevékenység legkorábbi kezdési időpontja a v-be vezető leghosszabb út hossza.

A PERT probléma

Egy $a, b, \ldots$ tevékenységekből álló projektet kell végrehajtanunk.
Precedenciafeltételek: bizonyos (u, v) párok esetén előírás, hogy az u tevékenységet a v előtt kell elvégezni, ezért v az u kezdetét követően $c(uv)$ időkorlát elteltével kezdhető.
Cél: minden v tevékenységehez olyan $k(v) \geq 0$ kezdési időpont meghatározása, ami nem sérti a preferenciafeltételeket, és a projekt végrehajtási ideje (vagyis a legnagyobb $k(v)$ érték) minimális.
G **irányított gráf** csúcsai a tevékenységek, élei pedig a precedenciafeltételek, az uv él hossza $c(uv)$.
Megf: (1) Ha G nem DAG, akkor a projekt nem hajtható végre.
(2) Ha G DAG, akkor minden v tevékenység legkorábbi kezdési időpontja a v-be vezető leghosszabb út hossza.
Köv: A PERT probléma megoldása nem más, mint a G DAG minden csúcsára az oda vezető leghosszabb út meghatározása.

A PERT probléma

Egy $a, b, \ldots$ tevékenységekből álló projektet kell végrehajtanunk.
Precedenciafeltételek: bizonyos (u, v) párok esetén előírás, hogy az u tevékenységet a v előtt kell elvégezni, ezért v az u kezdetét követően $c(uv)$ időkorlát elteltével kezdhető.
Cél: minden v tevékenységehez olyan $k(v) \geq 0$ kezdési időpont meghatározása, ami nem sérti a preferenciafeltételeket, és a projekt végrehajtási ideje (vagyis a legnagyobb $k(v)$ érték) minimális.
G **irányított gráf** csúcsai a tevékenységek, élei pedig a precedenciafeltételek, az uv él hossza $c(uv)$.
Megf: (1) Ha G nem DAG, akkor a projekt nem hajtható végre. (2) Ha G DAG, akkor minden v tevékenység legkorábbi kezdési időpontja a v-be vezető leghosszabb út hossza.
Köv: A PERT probléma megoldása nem más, mint a G DAG minden csúcsára az oda vezető leghosszabb út meghatározása.
Terminológia: G leghosszabb útja kritikus út, amiből több is lehet. Kritikus út csúcsai a kritikus tevékenységek.
Megf: Ha egy kritikus tevékenység nem kezdődik el a lehető legkorábbi időpontban, akkor az egész projekt végrehajtása csúszik.

Mit tanultunk ma?

Mit tanultunk ma?

- DFS: a BFS ,,ellentéte” (verem $\leftrightarrow$ FIFO)

Mit tanultunk ma?

- DFS: a BFS ,,ellentéte" (verem $\leftrightarrow$ FIFO)
- A DFS tulajdonságai, élek osztályozása a végpontok mélységi és befejezési számai alapján

Mit tanultunk ma?

- DFS: a BFS ,,ellentéte" (verem $\leftrightarrow$ FIFO)
- A DFS tulajdonságai, élek osztályozása a végpontok mélységi és befejezési számai alapján
- Irányítatlan DFS után nincs keresztél

Mit tanultunk ma?

- DFS: a BFS ,,ellentéte” (verem $\leftrightarrow$ FIFO)
- A DFS tulajdonságai, élek osztályozása a végpontok mélységi és befejezési számai alapján
- Irányítatlan DFS után nincs keresztél
- DAG, DFS visszaélek és topologikus sorrend kapcsolata

Mit tanultunk ma?

- DFS: a BFS ,,ellentéte" (verem $\leftrightarrow$ FIFO)
- A DFS tulajdonságai, élek osztályozása a végpontok mélységi és befejezési számai alapján
- Irányítatlan DFS után nincs keresztél
- DAG, DFS visszaélek és topologikus sorrend kapcsolata
- Leghosszabb út keresése DAG-ban

Mit tanultunk ma?

- DFS: a BFS ,,ellentéte" (verem $\leftrightarrow$ FIFO)
- A DFS tulajdonságai, élek osztályozása a végpontok mélységi és befejezési számai alapján
- Irányítatlan DFS után nincs keresztél
- DAG, DFS visszaélek és topologikus sorrend kapcsolata
- Leghosszabb út keresése DAG-ban
- Projektmenedzsment PERT-módszerrel

Mit tanultunk ma?

- DFS: a BFS ,,ellentéte” (verem $\leftrightarrow$ FIFO)
- A DFS tulajdonságai, élek osztályozása a végpontok mélységi és befejezési számai alapján
- Irányítatlan DFS után nincs keresztél
- DAG, DFS visszaélek és topologikus sorrend kapcsolata
- Leghosszabb út keresése DAG-ban
- Projektmenedzsment PERT-módszerrel

Köszönöm a figyelmet!