

A számítástudomány alapjai

Gráfbejárások és legrövidebb utak

2025. szeptember 30.

Munkaterv

Munkaterv

- ▶ Irányítatlan gráfok esetén a csúcsok közötti elérhetőséget a komponensek ill. feszítő erdő segítségével tudjuk kompakt módon leírni. Irányított gráfokban ez a struktúra ennél jóval bonyolultabb.

Munkaterv

- ▶ Irányítatlan gráfok esetén a csúcsok közötti elérhetőséget a komponensek ill. feszítő erdő segítségével tudjuk kompakt módon leírni. Irányított gráfokban ez a struktúra ennél jóval bonyolultabb.
- ▶ Irányított gráfban fogjuk azt vizsgálni, hogy egy megadott csúcsból a gráf milyen más csúcsai érhetőek el, majd az elérhető csúcsokba keresünk legrövidebb utat.

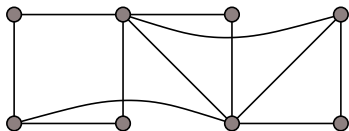
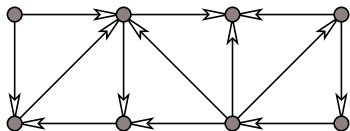
Munkaterv

- ▶ Irányítatlan gráfok esetén a csúcsok közötti elérhetőséget a komponensek ill. feszítő erdő segítségével tudjuk kompakt módon leírni. Irányított gráfokban ez a struktúra ennél jóval bonyolultabb.
- ▶ Irányított gráfban fogjuk azt vizsgálni, hogy egy megadott csúcsból a gráf milyen más csúcsai érhetők el, majd az elérhető csúcsokba keresünk legrövidebb utat.
- ▶ Ennek során (többek között) egy újfajta módszert látunk a feszítőfa (ill. feszítő erdő) keresésére.

Munkaterv

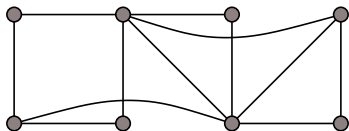
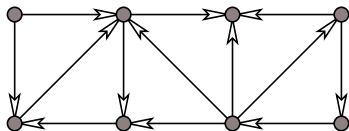
- ▶ Irányítatlan gráfok esetén a csúcsok közötti elérhetőséget a komponensek ill. feszítő erdő segítségével tudjuk kompakt módon leírni. Irányított gráfokban ez a struktúra ennél jóval bonyolultabb.
- ▶ Irányított gráfban fogjuk azt vizsgálni, hogy egy megadott csúcsból a gráf milyen más csúcsai érhetők el, majd az elérhető csúcsokba keresünk legrövidebb utat.
- ▶ Ennek során (többek között) egy újfajta módszert látunk a feszítőfa (ill. feszítő erdő) keresésére.
- ▶ Bár a továbbiakban irányított gráfokkal foglalkozunk, megállapításaink irányítatlan gráfokra is könnyen átültethetők. Ehhez az irányítatlan gráfból úgy képezünk irányított gráfot, hogy minden irányítatlan élt egy-egy oda-vissza irányított élpárral helyettesítünk.

Általános gráfbejárás



A gráfbejárási algoritmus az inputgráf csúcsait és éleit fedezi fel. Minden csúcs az **eléretlen** → **elért** → **befejezett** állapotokat veszi fel. A bejárás akkor ér véget, amint minden csúcs **befejezetté** vált.

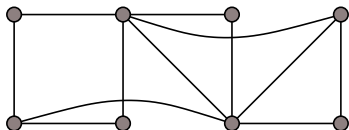
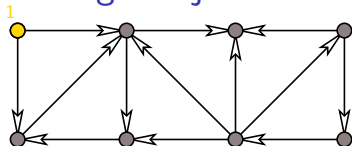
Általános gráfbejárás



A gráfbejárási algoritmus az inputgráf csúcsait és éleit fedezi fel. Minden csúcs az **eléretlen** $\rightarrow$ **elért** $\rightarrow$ **befejezett** állapotokat veszi fel. A bejárás akkor ér véget, amint minden csúcs **befejezetté** vált.

1. Van **elért** csúcs. Választunk egyet, mondjuk u -t.
 - (1a) Ha van olyan uv él, amire v eléretlen, akkor v **elértté** válik.
 - (1b) Ha nincs ilyen uv él, akkor u **befejezetté** válik.
2. Nincs **elért** csúcs.
 - (2a) Ha van eléretlen u csúcs, akkor u -t **elértté** tesszük.
 - (2b) Ha nincs eléretlen csúcs (azaz $\forall$ csúcs **befejezett**): END.

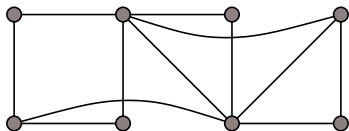
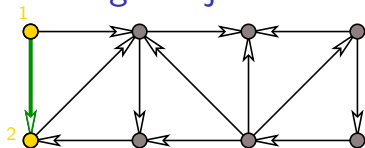
Általános gráfbejárás



A gráfbejárási algoritmus az inputgráf csúcsait és éleit fedezi fel. Minden csúcs az **eléretlen** $\rightarrow$ **elért** $\rightarrow$ **befejezett** állapotokat veszi fel. A bejárás akkor ér véget, amint minden csúcs **befejezetté** vált.

1. Van **elért** csúcs. Választunk egyet, mondjuk u -t.
 - (1a) Ha van olyan uv él, amire v eléretlen, akkor v **elértté** válik.
 - (1b) Ha nincs ilyen uv él, akkor u **befejezetté** válik.
2. Nincs **elért** csúcs.
 - (2a) Ha van eléretlen u csúcs, akkor u -t **elértté** tesszük.
 - (2b) Ha nincs eléretlen csúcs (azaz $\forall$ csúcs **befejezett**): END.

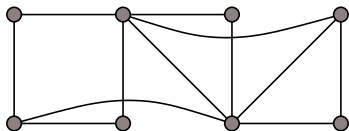
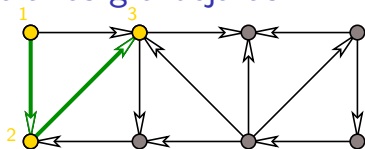
Általános gráfbejárás



A gráfbejárási algoritmus az inputgráf csúcsait és éleit fedezi fel. Minden csúcs az **eléretlen** $\rightarrow$ **elért** $\rightarrow$ **befejezett** állapotokat veszi fel. A bejárás akkor ér véget, amint minden csúcs **befejezett**té vált.

1. Van **elért** csúcs. Választunk egyet, mondjuk u -t.
 - (1a) Ha van olyan uv él, amire v **eléretlen**, akkor v **elért**té válik.
 - (1b) Ha nincs ilyen uv él, akkor u **befejezett**té válik.
2. Nincs **elért** csúcs.
 - (2a) Ha van **eléretlen** u csúcs, akkor u -t **elért**té tesszük.
 - (2b) Ha nincs **eléretlen** csúcs (azaz $\forall$ csúcs **befejezett**): END.

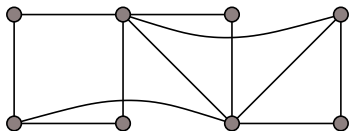
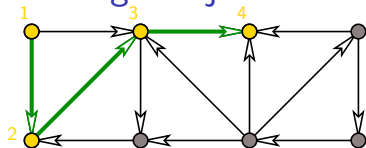
Általános gráfbejárás



A gráfbejárási algoritmus az inputgráf csúcsait és éleit fedezi fel. Minden csúcs az **eléretlen** → **elért** → **befejezett** állapotokat veszi fel. A bejárás akkor ér véget, amint minden csúcs **befejezetté** vált.

1. Van **elért** csúcs. Választunk egyet, mondjuk u -t.
 - (1a) Ha van olyan uv él, amire v eléretlen, akkor v **elértté** válik.
 - (1b) Ha nincs ilyen uv él, akkor u **befejezetté** válik.
2. Nincs **elért** csúcs.
 - (2a) Ha van eléretlen u csúcs, akkor u -t **elértté** tesszük.
 - (2b) Ha nincs eléretlen csúcs (azaz $\forall$ csúcs **befejezett**): END.

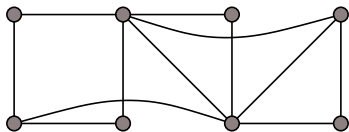
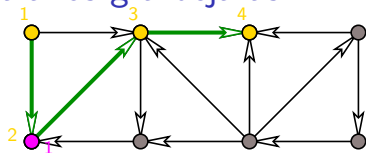
Általános gráfbejárás



A gráfbejárási algoritmus az inputgráf csúcsait és éleit fedezi fel. Minden csúcs az **eléretlen** $\rightarrow$ **elért** $\rightarrow$ **befejezett** állapotokat veszi fel. A bejárás akkor ér véget, amint minden csúcs **befejezetté** vált.

1. Van **elért** csúcs. Választunk egyet, mondjuk u -t.
 - (1a) Ha van olyan uv él, amire v eléretlen, akkor v **elértté** válik.
 - (1b) Ha nincs ilyen uv él, akkor u **befejezetté** válik.
2. Nincs **elért** csúcs.
 - (2a) Ha van eléretlen u csúcs, akkor u -t **elértté** tesszük.
 - (2b) Ha nincs eléretlen csúcs (azaz $\forall$ csúcs **befejezett**): END.

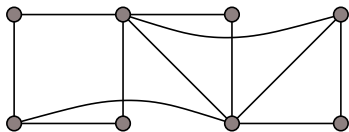
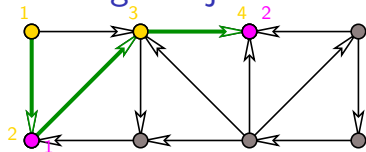
Általános gráfbejárás



A gráfbejárási algoritmus az inputgráf csúcsait és éleit fedezi fel. Minden csúcs az **eléretlen** $\rightarrow$ **elért** $\rightarrow$ **befejezett** állapotokat veszi fel. A bejárás akkor ér véget, amint minden csúcs **befejezett**té vált.

1. Van **elért** csúcs. Választunk egyet, mondjuk u -t.
 - (1a) Ha van olyan uv él, amire v eléretlen, akkor v **elért**té válik.
 - (1b) Ha nincs ilyen uv él, akkor u **befejezett**té válik.
2. Nincs **elért** csúcs.
 - (2a) Ha van eléretlen u csúcs, akkor u -t **elért**té tesszük.
 - (2b) Ha nincs eléretlen csúcs (azaz $\forall$ csúcs **befejezett**): END.

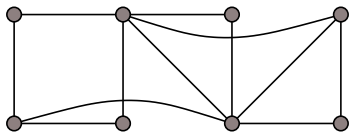
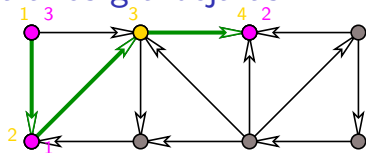
Általános gráfbejárás



A gráfbejárási algoritmus az inputgráf csúcsait és éleit fedezi fel. Minden csúcs az **eléretlen** $\rightarrow$ **elért** $\rightarrow$ **befejezett** állapotokat veszi fel. A bejárás akkor ér véget, amint minden csúcs **befejezett**té vált.

1. Van **elért** csúcs. Választunk egyet, mondjuk u -t.
 - (1a) Ha van olyan uv él, amire v eléretlen, akkor v **elért**té válik.
 - (1b) Ha nincs ilyen uv él, akkor u **befejezett**té válik.
2. Nincs **elért** csúcs.
 - (2a) Ha van eléretlen u csúcs, akkor u -t **elért**té tesszük.
 - (2b) Ha nincs eléretlen csúcs (azaz $\forall$ csúcs **befejezett**): END.

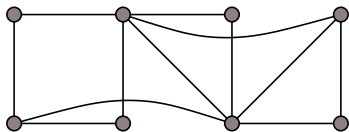
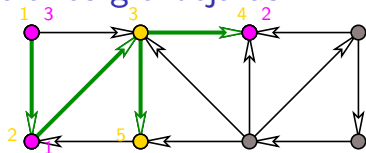
Általános gráfbejárás



A gráfbejárási algoritmus az inputgráf csúcsait és éleit fedezi fel. Minden csúcs az **eléretlen** $\rightarrow$ **elért** $\rightarrow$ **befejezett** állapotokat veszi fel. A bejárás akkor ér véget, amint minden csúcs **befejezett**té vált.

1. Van **elért** csúcs. Választunk egyet, mondjuk u -t.
 - (1a) Ha van olyan uv él, amire v eléretlen, akkor v **elért**té válik.
 - (1b) Ha nincs ilyen uv él, akkor u **befejezett**té válik.
2. Nincs **elért** csúcs.
 - (2a) Ha van eléretlen u csúcs, akkor u -t **elért**té tesszük.
 - (2b) Ha nincs eléretlen csúcs (azaz $\forall$ csúcs **befejezett**): END.

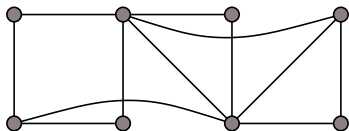
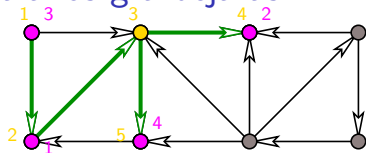
Általános gráfbejárás



A gráfbejárési algoritmus az inputgráf csúcsait és éleit fedezi fel. Minden csúcs az **eléretlen** $\rightarrow$ **elért** $\rightarrow$ **befejezett** állapotokat veszi fel. A bejárás akkor ér véget, amint minden csúcs **befejezett**té vált.

1. Van **elért** csúcs. Választunk egyet, mondjuk u -t.
 - (1a) Ha van olyan uv él, amire v eléretlen, akkor v **elért**té válik.
 - (1b) Ha nincs ilyen uv él, akkor u **befejezett**té válik.
2. Nincs **elért** csúcs.
 - (2a) Ha van eléretlen u csúcs, akkor u -t **elért**té tesszük.
 - (2b) Ha nincs eléretlen csúcs (azaz $\forall$ csúcs **befejezett**): END.

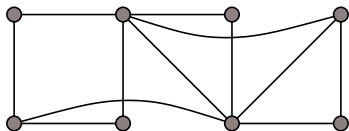
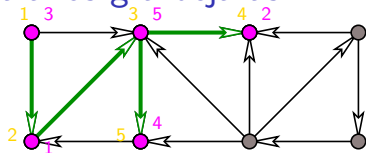
Általános gráfbejárás



A gráfbejárési algoritmus az inputgráf csúcsait és éleit fedezi fel. Minden csúcs az **eléretlen** $\rightarrow$ **elért** $\rightarrow$ **befejezett** állapotokat veszi fel. A bejárás akkor ér véget, amint minden csúcs **befejezett**té vált.

1. Van **elért** csúcs. Választunk egyet, mondjuk u -t.
 - (1a) Ha van olyan uv él, amire v eléretlen, akkor v **elért**té válik.
 - (1b) Ha nincs ilyen uv él, akkor u **befejezett**té válik.
2. Nincs **elért** csúcs.
 - (2a) Ha van eléretlen u csúcs, akkor u -t **elért**té tesszük.
 - (2b) Ha nincs eléretlen csúcs (azaz $\forall$ csúcs **befejezett**): END.

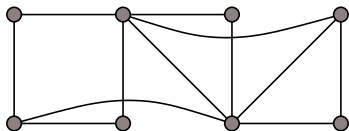
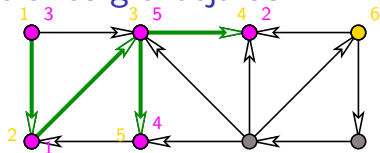
Általános gráfbejárás



A gráfbejárási algoritmus az inputgráf csúcsait és éleit fedezi fel. Minden csúcs az **eléretlen** $\rightarrow$ **elért** $\rightarrow$ **befejezett** állapotokat veszi fel. A bejárás akkor ér véget, amint minden csúcs **befejezett**té vált.

1. Van **elért** csúcs. Választunk egyet, mondjuk u -t.
 - (1a) Ha van olyan uv él, amire v eléretlen, akkor v **elért**té válik.
 - (1b) Ha nincs ilyen uv él, akkor u **befejezett**té válik.
2. Nincs **elért** csúcs.
 - (2a) Ha van eléretlen u csúcs, akkor u -t **elért**té tesszük.
 - (2b) Ha nincs eléretlen csúcs (azaz $\forall$ csúcs **befejezett**): END.

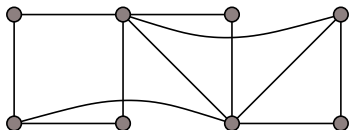
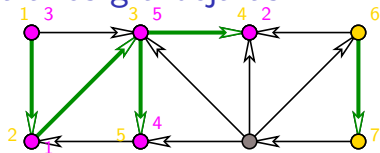
Általános gráfbejárás



A gráfbejárási algoritmus az inputgráf csúcsait és éleit fedezi fel. Minden csúcs az **eléretlen** $\rightarrow$ **elért** $\rightarrow$ **befejezett** állapotokat veszi fel. A bejárás akkor ér véget, amint minden csúcs **befejezett**té vált.

1. Van **elért** csúcs. Választunk egyet, mondjuk u -t.
 - (1a) Ha van olyan uv él, amire v eléretlen, akkor v **elért**té válik.
 - (1b) Ha nincs ilyen uv él, akkor u **befejezett**té válik.
2. Nincs **elért** csúcs.
 - (2a) Ha van eléretlen u csúcs, akkor u -t **elért**té tesszük.
 - (2b) Ha nincs eléretlen csúcs (azaz $\forall$ csúcs **befejezett**): END.

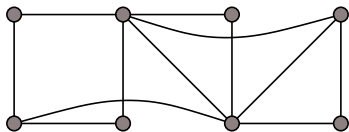
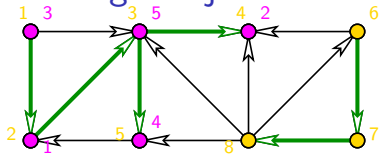
Általános gráfbejárás



A gráfbejárási algoritmus az inputgráf csúcsait és éleit fedezi fel. Minden csúcs az **eléretlen** → **elért** → **befejezett** állapotokat veszi fel. A bejárás akkor ér véget, amint minden csúcs **befejezetté** vált.

1. Van **elért** csúcs. Választunk egyet, mondjuk u -t.
 - (1a) Ha van olyan uv él, amire v eléretlen, akkor v **elértté** válik.
 - (1b) Ha nincs ilyen uv él, akkor u **befejezetté** válik.
2. Nincs **elért** csúcs.
 - (2a) Ha van eléretlen u csúcs, akkor u -t **elértté** tesszük.
 - (2b) Ha nincs eléretlen csúcs (azaz $\forall$ csúcs **befejezett**): END.

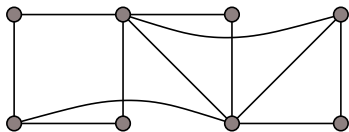
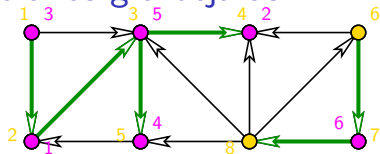
Általános gráfbejárás



A gráfbejárási algoritmus az inputgráf csúcsait és éleit fedezi fel. Minden csúcs az **eléretlen** → **elért** → **befejezett** állapotokat veszi fel. A bejárás akkor ér véget, amint minden csúcs **befejezetté** vált.

1. Van **elért** csúcs. Választunk egyet, mondjuk u -t.
 - (1a) Ha van olyan uv él, amire v eléretlen, akkor v **elértté** válik.
 - (1b) Ha nincs ilyen uv él, akkor u **befejezetté** válik.
2. Nincs **elért** csúcs.
 - (2a) Ha van eléretlen u csúcs, akkor u -t **elértté** tesszük.
 - (2b) Ha nincs eléretlen csúcs (azaz $\forall$ csúcs **befejezett**): END.

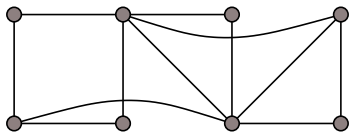
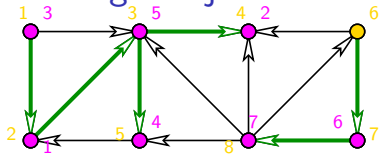
Általános gráfbejárás



A gráfbejárási algoritmus az inputgráf csúcsait és éleit fedezi fel. Minden csúcs az **eléretlen** $\rightarrow$ **elért** $\rightarrow$ **befejezett** állapotokat veszi fel. A bejárás akkor ér véget, amint minden csúcs **befejezett**té vált.

1. Van **elért** csúcs. Választunk egyet, mondjuk u -t.
 - (1a) Ha van olyan uv él, amire v eléretlen, akkor v **elért**té válik.
 - (1b) Ha nincs ilyen uv él, akkor u **befejezett**té válik.
2. Nincs **elért** csúcs.
 - (2a) Ha van eléretlen u csúcs, akkor u -t **elért**té tesszük.
 - (2b) Ha nincs eléretlen csúcs (azaz $\forall$ csúcs **befejezett**): END.

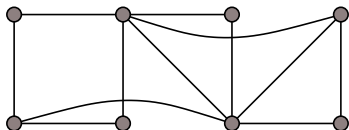
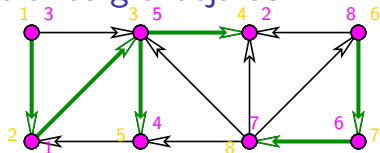
Általános gráfbejárás



A gráfbejárási algoritmus az inputgráf csúcsait és éleit fedezi fel. Minden csúcs az **eléretlen** → **elért** → **befejezett** állapotokat veszi fel. A bejárás akkor ér véget, amint minden csúcs **befejezetté** vált.

1. Van **elért** csúcs. Választunk egyet, mondjuk u -t.
 - (1a) Ha van olyan uv él, amire v eléretlen, akkor v **elértté** válik.
 - (1b) Ha nincs ilyen uv él, akkor u **befejezetté** válik.
2. Nincs **elért** csúcs.
 - (2a) Ha van eléretlen u csúcs, akkor u -t **elértté** tesszük.
 - (2b) Ha nincs eléretlen csúcs (azaz $\forall$ csúcs **befejezett**): END.

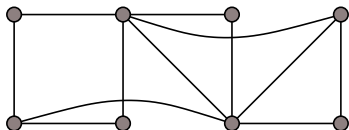
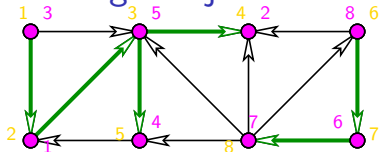
Általános gráfbejárás



A gráfbejárási algoritmus az inputgráf csúcsait és éleit fedezi fel. Minden csúcs az eléretlen $\rightarrow$ **elért** $\rightarrow$ **befejezett** állapotokat veszi fel. A bejárás akkor ér véget, amint minden csúcs **befejezett**té vált.

1. Van **elért** csúcs. Választunk egyet, mondjuk u -t.
 - (1a) Ha van olyan uv él, amire v eléretlen, akkor v **elért**té válik.
 - (1b) Ha nincs ilyen uv él, akkor u **befejezett**té válik.
2. Nincs **elért** csúcs.
 - (2a) Ha van eléretlen u csúcs, akkor u -t **elért**té tesszük.
 - (2b) Ha nincs eléretlen csúcs (azaz $\forall$ csúcs **befejezett**): END.

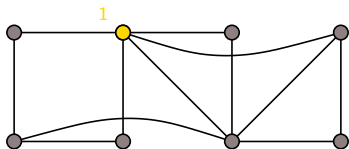
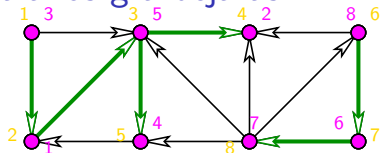
Általános gráfbejárás



A gráfbejárási algoritmus az inputgráf csúcsait és éleit fedezi fel. Minden csúcs az **eléretlen** → **elért** → **befejezett** állapotokat veszi fel. A bejárás akkor ér véget, amint minden csúcs **befejezetté** vált.

1. Van **elért** csúcs. Választunk egyet, mondjuk u -t.
 - (1a) Ha van olyan uv él, amire v **eléletlen**, akkor v **elértté** válik.
 - (1b) Ha nincs ilyen uv él, akkor u **befejezetté** válik.
 2. Nincs **elért** csúcs.
 - (2a) Ha van **eléletlen** u csúcs, akkor u -t **elértté** tesszük.
 - (2b) Ha nincs **eléletlen** csúcs (azaz $\forall$ csúcs **befejezett**): END.
- Nézzük meg egy irányítatlan gráf bejárását is.

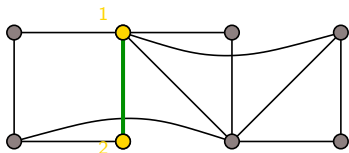
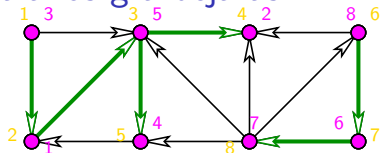
Általános gráfbejárás



A gráfbejárési algoritmus az inputgráf csúcsait és éleit fedezi fel. Minden csúcs az **eléretlen** $\rightarrow$ **elért** $\rightarrow$ **befejezett** állapotokat veszi fel. A bejárás akkor ér véget, amint minden csúcs **befejezett**té vált.

1. Van **elért** csúcs. Választunk egyet, mondjuk u -t.
 - (1a) Ha van olyan uv él, amire v eléretlen, akkor v **elért**té válik.
 - (1b) Ha nincs ilyen uv él, akkor u **befejezett**té válik.
 2. Nincs **elért** csúcs.
 - (2a) Ha van eléretlen u csúcs, akkor u -t **elért**té tesszük.
 - (2b) Ha nincs eléretlen csúcs (azaz $\forall$ csúcs **befejezett**): END.
- Nézzük meg egy irányítatlan gráf bejárását is.

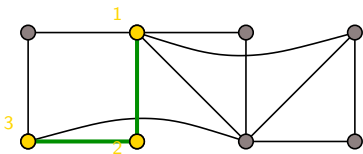
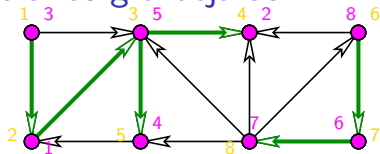
Általános gráfbejárás



A gráfbejárési algoritmus az inputgráf csúcsait és éleit fedezi fel. Minden csúcs az elértlen $\rightarrow$ **elért** $\rightarrow$ **befejezett** állapotokat veszi fel. A bejárás akkor ér véget, amint minden csúcs **befejezett**té vált.

1. Van **elért** csúcs. Választunk egyet, mondjuk u -t.
 - (1a) Ha van olyan uv él, amire v elértlen, akkor v **elért**té válik.
 - (1b) Ha nincs ilyen uv él, akkor u **befejezett**té válik.
 2. Nincs **elért** csúcs.
 - (2a) Ha van elértlen u csúcs, akkor u -t **elért**té tesszük.
 - (2b) Ha nincs elértlen csúcs (azaz $\forall$ csúcs **befejezett**): END.
- Nézzük meg egy irányítatlan gráf bejárását is.

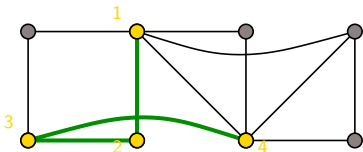
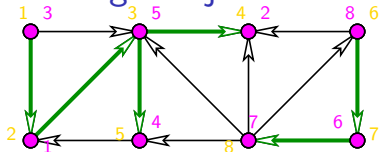
Általános gráfbejárás



A gráfbejárási algoritmus az inputgráf csúcsait és éleit fedezi fel. Minden csúcs az **eléretlen** $\rightarrow$ **elért** $\rightarrow$ **befejezett** állapotokat veszi fel. A bejárás akkor ér véget, amint minden csúcs **befejezett**té vált.

1. Van **elért** csúcs. Választunk egyet, mondjuk u -t.
 - (1a) Ha van olyan uv él, amire v eléretlen, akkor v **elért**té válik.
 - (1b) Ha nincs ilyen uv él, akkor u **befejezett**té válik.
 2. Nincs **elért** csúcs.
 - (2a) Ha van eléretlen u csúcs, akkor u -t **elért**té tesszük.
 - (2b) Ha nincs eléretlen csúcs (azaz $\forall$ csúcs **befejezett**): END.
- Nézzük meg egy irányítatlan gráf bejárását is.

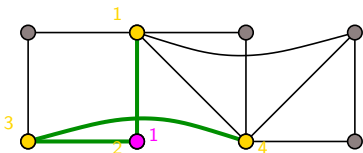
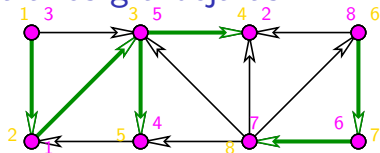
Általános gráfbejárás



A gráfbejárási algoritmus az inputgráf csúcsait és éleit fedezi fel. Minden csúcs az **eléretlen** $\rightarrow$ **elért** $\rightarrow$ **befejezett** állapotokat veszi fel. A bejárás akkor ér véget, amint minden csúcs **befejezetté** vált.

1. Van **elért** csúcs. Választunk egyet, mondjuk u -t.
 - (1a) Ha van olyan uv él, amire v **eléletlen**, akkor v **elértté** válik.
 - (1b) Ha nincs ilyen uv él, akkor u **befejezetté** válik.
 2. Nincs **elért** csúcs.
 - (2a) Ha van **eléletlen** u csúcs, akkor u -t **elértté** tesszük.
 - (2b) Ha nincs **eléletlen** csúcs (azaz $\forall$ csúcs **befejezett**): END.
- Nézzük meg egy irányítatlan gráf bejárását is.

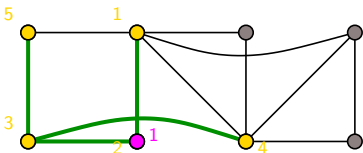
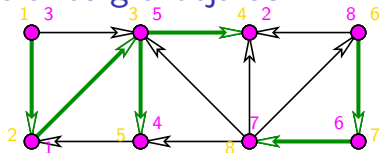
Általános gráfbejárás



A gráfbejárási algoritmus az inputgráf csúcsait és éleit fedezi fel. Minden csúcs az **eléretlen** $\rightarrow$ **elért** $\rightarrow$ **befejezett** állapotokat veszi fel. A bejárás akkor ér véget, amint minden csúcs **befejezett**té vált.

1. Van **elért** csúcs. Választunk egyet, mondjuk u -t.
 - (1a) Ha van olyan uv él, amire v eléretlen, akkor v **elért**té válik.
 - (1b) Ha nincs ilyen uv él, akkor u **befejezett**té válik.
 2. Nincs **elért** csúcs.
 - (2a) Ha van eléretlen u csúcs, akkor u -t **elért**té tesszük.
 - (2b) Ha nincs eléretlen csúcs (azaz $\forall$ csúcs **befejezett**): END.
- Nézzük meg egy irányítatlan gráf bejárását is.

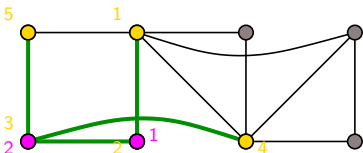
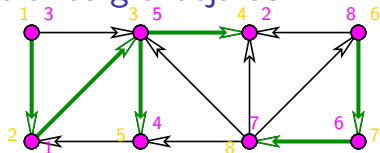
Általános gráfbejárás



A gráfbejárási algoritmus az inputgráf csúcsait és éleit fedezi fel. Minden csúcs az **eléretlen** $\rightarrow$ **elért** $\rightarrow$ **befejezett** állapotokat veszi fel. A bejárás akkor ér véget, amint minden csúcs **befejezett**té vált.

1. Van **elért** csúcs. Választunk egyet, mondjuk u -t.
 - (1a) Ha van olyan uv él, amire v eléretlen, akkor v **elért**té válik.
 - (1b) Ha nincs ilyen uv él, akkor u **befejezett**té válik.
 2. Nincs **elért** csúcs.
 - (2a) Ha van eléretlen u csúcs, akkor u -t **elért**té tesszük.
 - (2b) Ha nincs eléretlen csúcs (azaz $\forall$ csúcs **befejezett**): END.
- Nézzük meg egy irányítatlan gráf bejárását is.

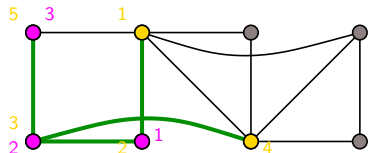
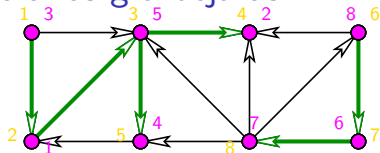
Általános gráfbejárás



A gráfbejárási algoritmus az inputgráf csúcsait és éleit fedezi fel. Minden csúcs az **eléretlen** → **elért** → **befejezett** állapotokat veszi fel. A bejárás akkor ér véget, amint minden csúcs **befejezetté** vált.

1. Van **elért** csúcs. Választunk egyet, mondjuk u -t.
 - (1a) Ha van olyan uv él, amire v **eléletlen**, akkor v **elértté** válik.
 - (1b) Ha nincs ilyen uv él, akkor u **befejezetté** válik.
 2. Nincs **elért** csúcs.
 - (2a) Ha van **eléletlen** u csúcs, akkor u -t **elértté** tesszük.
 - (2b) Ha nincs **eléletlen** csúcs (azaz $\forall$ csúcs **befejezett**): END.
- Nézzük meg egy irányítatlan gráf bejárását is.

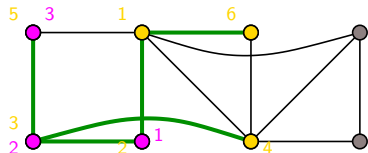
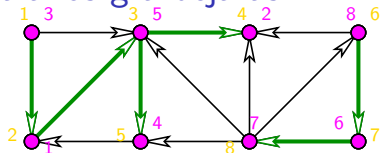
Általános gráfbejárás



A gráfbejárási algoritmus az inputgráf csúcsait és éleit fedezi fel. Minden csúcs az **eléretlen** $\rightarrow$ **elért** $\rightarrow$ **befejezett** állapotokat veszi fel. A bejárás akkor ér véget, amint minden csúcs **befejezett**té vált.

1. Van **elért** csúcs. Választunk egyet, mondjuk u -t.
 - (1a) Ha van olyan uv él, amire v eléretlen, akkor v **elért**té válik.
 - (1b) Ha nincs ilyen uv él, akkor u **befejezett**té válik.
 2. Nincs **elért** csúcs.
 - (2a) Ha van eléretlen u csúcs, akkor u -t **elért**té tesszük.
 - (2b) Ha nincs eléretlen csúcs (azaz $\forall$ csúcs **befejezett**): END.
- Nézzük meg egy irányítatlan gráf bejárását is.

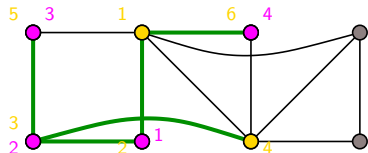
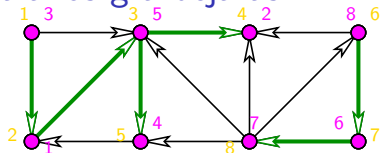
Általános gráfbejárás



A gráfbejárási algoritmus az inputgráf csúcsait és éleit fedezi fel. Minden csúcs az **eléretlen** $\rightarrow$ **elért** $\rightarrow$ **befejezett** állapotokat veszi fel. A bejárás akkor ér véget, amint minden csúcs **befejezett**té vált.

1. Van **elért** csúcs. Választunk egyet, mondjuk u -t.
 - (1a) Ha van olyan uv él, amire v eléretlen, akkor v **elért**té válik.
 - (1b) Ha nincs ilyen uv él, akkor u **befejezett**té válik.
 2. Nincs **elért** csúcs.
 - (2a) Ha van eléretlen u csúcs, akkor u -t **elért**té tesszük.
 - (2b) Ha nincs eléretlen csúcs (azaz $\forall$ csúcs **befejezett**): END.
- Nézzük meg egy irányítatlan gráf bejárását is.

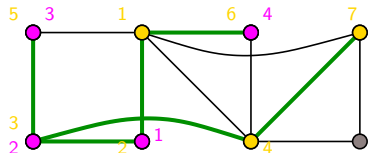
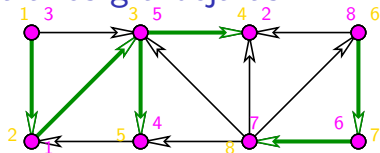
Általános gráfbejárás



A gráfbejárési algoritmus az inputgráf csúcsait és éleit fedezi fel. Minden csúcs az eléretlen $\rightarrow$ **elért** $\rightarrow$ **befejezett** állapotokat veszi fel. A bejárás akkor ér véget, amint minden csúcs **befejezett**té vált.

1. Van **elért** csúcs. Választunk egyet, mondjuk u -t.
 - (1a) Ha van olyan uv él, amire v eléretlen, akkor v **elért**té válik.
 - (1b) Ha nincs ilyen uv él, akkor u **befejezett**té válik.
 2. Nincs **elért** csúcs.
 - (2a) Ha van eléretlen u csúcs, akkor u -t **elért**té tesszük.
 - (2b) Ha nincs eléretlen csúcs (azaz $\forall$ csúcs **befejezett**): END.
- Nézzük meg egy irányítatlan gráf bejárását is.

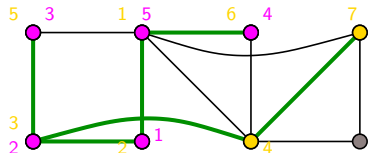
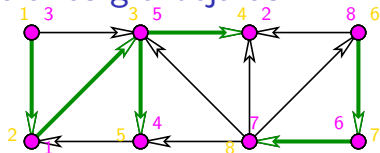
Általános gráfbejárás



A gráfbejárási algoritmus az inputgráf csúcsait és éleit fedezi fel. Minden csúcs az **eléretlen** $\rightarrow$ **elért** $\rightarrow$ **befejezett** állapotokat veszi fel. A bejárás akkor ér véget, amint minden csúcs **befejezetté** vált.

1. Van **elért** csúcs. Választunk egyet, mondjuk u -t.
 - (1a) Ha van olyan uv él, amire v eléretlen, akkor v **elértté** válik.
 - (1b) Ha nincs ilyen uv él, akkor u **befejezetté** válik.
 2. Nincs **elért** csúcs.
 - (2a) Ha van eléretlen u csúcs, akkor u -t **elértté** tesszük.
 - (2b) Ha nincs eléretlen csúcs (azaz $\forall$ csúcs **befejezett**): END.
- Nézzük meg egy irányítatlan gráf bejárását is.

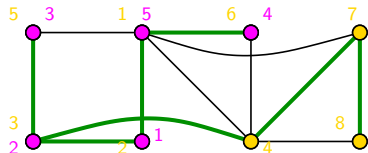
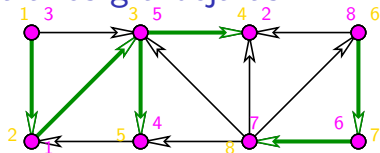
Általános gráfbejárás



A gráfbejárási algoritmus az inputgráf csúcsait és éleit fedezi fel. Minden csúcs az **eléretlen** $\rightarrow$ **elért** $\rightarrow$ **befejezett** állapotokat veszi fel. A bejárás akkor ér véget, amint minden csúcs **befejezetté** vált.

1. Van **elért** csúcs. Választunk egyet, mondjuk u -t.
 - (1a) Ha van olyan uv él, amire v eléretlen, akkor v **elértté** válik.
 - (1b) Ha nincs ilyen uv él, akkor u **befejezetté** válik.
 2. Nincs **elért** csúcs.
 - (2a) Ha van eléretlen u csúcs, akkor u -t **elértté** tesszük.
 - (2b) Ha nincs eléretlen csúcs (azaz $\forall$ csúcs **befejezett**): END.
- Nézzük meg egy irányítatlan gráf bejárását is.

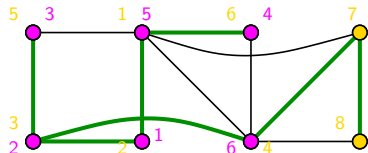
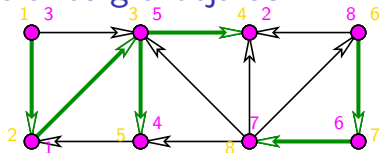
Általános gráfbejárás



A gráfbejárási algoritmus az inputgráf csúcsait és éleit fedezi fel. Minden csúcs az **eléretlen** $\rightarrow$ **elért** $\rightarrow$ **befejezett** állapotokat veszi fel. A bejárás akkor ér véget, amint minden csúcs **befejezetté** vált.

1. Van **elért** csúcs. Választunk egyet, mondjuk u -t.
 - (1a) Ha van olyan uv él, amire v eléretlen, akkor v **elértté** válik.
 - (1b) Ha nincs ilyen uv él, akkor u **befejezetté** válik.
 2. Nincs **elért** csúcs.
 - (2a) Ha van eléretlen u csúcs, akkor u -t **elértté** tesszük.
 - (2b) Ha nincs eléretlen csúcs (azaz $\forall$ csúcs **befejezett**): END.
- Nézzük meg egy irányítatlan gráf bejárását is.

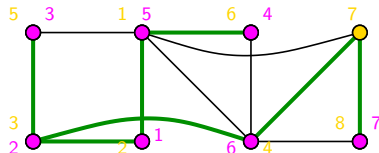
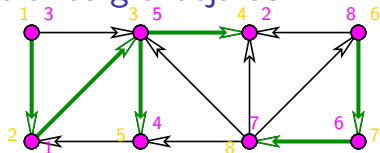
Általános gráfbejárás



A gráfbejárési algoritmus az inputgráf csúcsait és éleit fedezi fel. Minden csúcs az **eléretlen** $\rightarrow$ **elért** $\rightarrow$ **befejezett** állapotokat veszi fel. A bejárás akkor ér véget, amint minden csúcs **befejezett**té vált.

1. Van **elért** csúcs. Választunk egyet, mondjuk u -t.
 - (1a) Ha van olyan uv él, amire v eléretlen, akkor v **elért**té válik.
 - (1b) Ha nincs ilyen uv él, akkor u **befejezett**té válik.
 2. Nincs **elért** csúcs.
 - (2a) Ha van eléretlen u csúcs, akkor u -t **elért**té tesszük.
 - (2b) Ha nincs eléretlen csúcs (azaz $\forall$ csúcs **befejezett**): END.
- Nézzük meg egy irányítatlan gráf bejárását is.

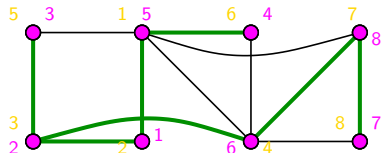
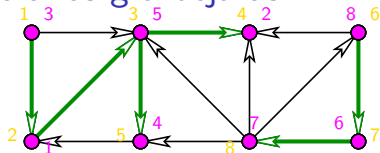
Általános gráfbejárás



A gráfbejárési algoritmus az inputgráf csúcsait és éleit fedezi fel. Minden csúcs az **eléretlen** $\rightarrow$ **elért** $\rightarrow$ **befejezett** állapotokat veszi fel. A bejárás akkor ér véget, amint minden csúcs **befejezetté** vált.

1. Van **elért** csúcs. Választunk egyet, mondjuk u -t.
 - (1a) Ha van olyan uv él, amire v eléretlen, akkor v **elértté** válik.
 - (1b) Ha nincs ilyen uv él, akkor u **befejezetté** válik.
 2. Nincs **elért** csúcs.
 - (2a) Ha van eléretlen u csúcs, akkor u -t **elértté** tesszük.
 - (2b) Ha nincs eléretlen csúcs (azaz $\forall$ csúcs **befejezett**): END.
- Nézzük meg egy irányítatlan gráf bejárását is.

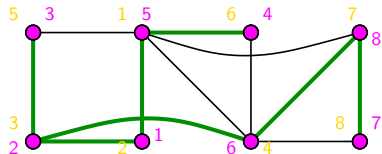
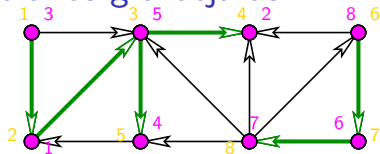
Általános gráfbejárás



A gráfbejárési algoritmus az inputgráf csúcsait és éleit fedezi fel. Minden csúcs az **eléretlen** $\rightarrow$ **elért** $\rightarrow$ **befejezett** állapotokat veszi fel. A bejárás akkor ér véget, amint minden csúcs **befejezetté** vált.

1. Van **elért** csúcs. Választunk egyet, mondjuk u -t.
 - (1a) Ha van olyan uv él, amire v eléretlen, akkor v **elértté** válik.
 - (1b) Ha nincs ilyen uv él, akkor u **befejezetté** válik.
 2. Nincs **elért** csúcs.
 - (2a) Ha van eléretlen u csúcs, akkor u -t **elértté** tesszük.
 - (2b) Ha nincs eléretlen csúcs (azaz $\forall$ csúcs **befejezett**): END.
- Nézzük meg egy irányítatlan gráf bejárását is.

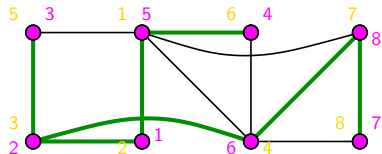
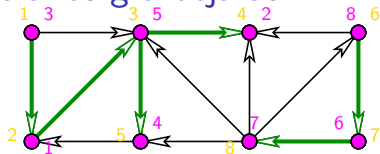
Általános gráfbejárás



Input: $G = (V, E)$ (ir/ir.tatlan) gráf, (esetleg $r \in V$ gyökér¹).

¹A gyökér már kezdetben **elért** állapotú, így kivétel az általános szabály alól.

Általános gráfbejárás

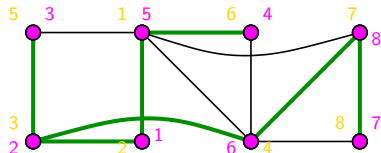
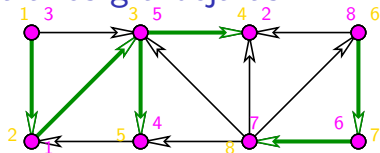


Input: $G = (V, E)$ (ir/ir.tatlan) gráf, (esetleg $r \in V$ gyökér¹).

Output: (1) A csúcsok **elérési** és **befejezési** sorrendje.

¹A gyökér már kezdetben **elért** állapotú, így kivétel az általános szabály alól.

Általános gráfbejárás



Input: $G = (V, E)$ (ir/ir.tatlan) gráf, (esetleg $r \in V$ gyökér¹).

Output: (1) A csúcsok **elérési** és **befejezési** sorrendje.

(2) Az élek osztályozása:

faél: Olyan él, ami mentén egy csúcs elértté vált.

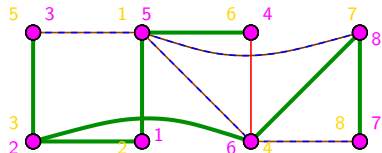
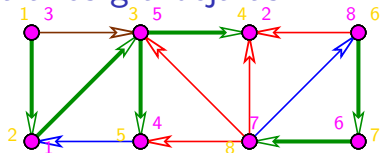
uv **előreél:** nem faél, de u -ból v -be faélekből irányított út vezet.

uv **visszaél:** v -ből u -ba faélekből irányított út vezet.

keresztél: minden más él (u és v közt nincs leszármazási viszony).

¹A gyökér már kezdetben **elért** állapotú, így kivétel az általános szabály alól.

Általános gráfbejárás



Input: $G = (V, E)$ (ir/ir.tatlan) gráf, (esetleg $r \in V$ gyökér¹).

Output: (1) A csúcsok **elérési** és **befejezési** sorrendje.

(2) Az élek osztályozása:

faél: Olyan él, ami mentén egy csúcs elértté vált.

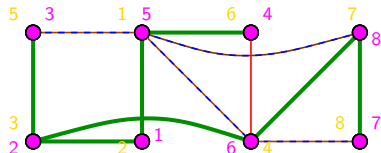
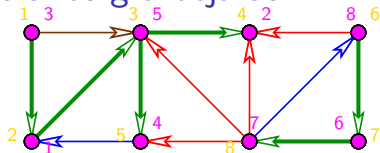
uv **előreél:** nem faél, de u -ból v -be faélekből irányított út vezet.

uv **visszaél:** v -ből u -ba faélekből irányított út vezet.

keresztél: minden más él (u és v közt nincs leszármazási viszony).

¹A gyökér már kezdetben **elért** állapotú, így kivétel az általános szabály alól.

Általános gráfbejárás



Input: $G = (V, E)$ (ir/ir.tatlan) gráf, (esetleg $r \in V$ gyökér¹).

Output: (1) A csúcsok **elérési** és **befejezési** sorrendje.

(2) Az élek osztályozása:

faél: Olyan él, ami mentén egy csúcs elértté vált.

uv **előreél:** nem faél, de u -ból v -be faélekből irányított út vezet.

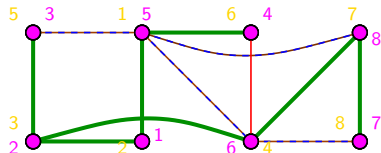
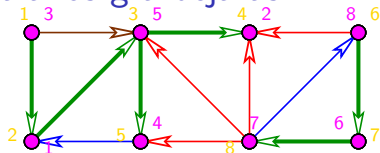
uv **visszaél:** v -ből u -ba faélekből irányított út vezet.

keresztél: minden más él (u és v közt nincs leszármazási viszony).

(3) A **bejárás fája:** a faélek alkotta részgráf.

¹A gyökér már kezdetben **elért** állapotú, így kivétel az általános szabály alól.

Általános gráfbejárás



Input: $G = (V, E)$ (ir/ir.tatlan) gráf, (esetleg $r \in V$ gyökér¹).

Output: (1) A csúcsok **elérési** és **befejezési** sorrendje.

(2) Az élek osztályozása:

faél: Olyan él, ami mentén egy csúcs elértté vált.

uv **előreél:** nem faél, de u -ból v -be faélekből irányított út vezet.

uv **visszaél:** v -ből u -ba faélekből irányított út vezet.

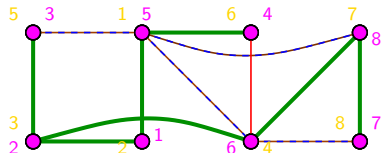
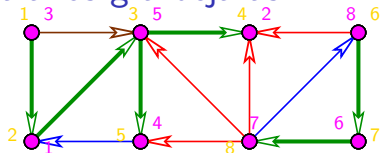
keresztél: minden más él (u és v közt nincs leszármazási viszony).

(3) A **bejárás fája**: a faélek alkotta részgráf.

(A bejárás fája valójában egy gyökereiből kifelé irányított erdő.)

¹A gyökér már kezdetben **elért** állapotú, így kivétel az általános szabály alól.

Általános gráfbejárás



Input: $G = (V, E)$ (ir/ir.tatlan) gráf, (esetleg $r \in V$ gyökér¹).

Output: (1) A csúcsok **elérési** és **befejezési** sorrendje.

(2) Az élek osztályozása:

faél: Olyan él, ami mentén egy csúcs elértté vált.

uv **előreél:** nem faél, de u -ból v -be faélekből irányított út vezet.

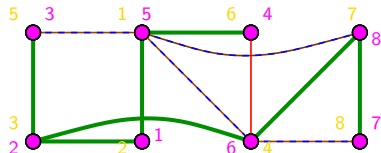
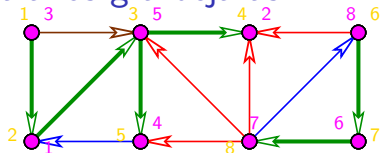
uv **visszaél:** v -ből u -ba faélekből irányított út vezet.

keresztél: minden más él (u és v közt nincs leszármazási viszony).

(3) A **bejárás fája**: a faélek alkotta részgráf.

¹A gyökér már kezdetben **elért** állapotú, így kivétel az általános szabály alól.

Általános gráfbejárás



Input: $G = (V, E)$ (ir/ir.tatlan) gráf, (esetleg $r \in V$ gyökér¹).

Output: (1) A csúcsok **elérési** és **befejezési** sorrendje.

(2) Az élek osztályozása:

faél: Olyan él, ami mentén egy csúcs elértté vált.

uv **előreél:** nem faél, de u -ból v -be faélekből irányított út vezet.

uv **visszaél:** v -ből u -ba faélekből irányított út vezet.

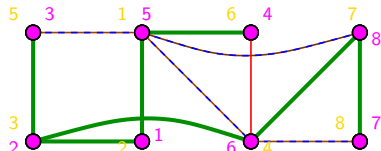
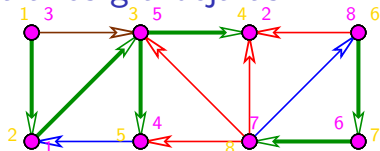
keresztél: minden más él (u és v közt nincs leszármazási viszony).

(3) A **bejárás fája:** a faélek alkotta részgráf.

Megf: Irányítatlan esetben az előreél és a visszaél ugyanazt jelenti.

¹A gyökér már kezdetben **elért** állapotú, így kivétel az általános szabály alól.

Általános gráfbejárás



Input: $G = (V, E)$ (ir/ir.tatlan) gráf, (esetleg $r \in V$ gyökér¹).

Output: (1) A csúcsok **elérési** és **befejezési** sorrendje.

(2) Az élek osztályozása:

faél: Olyan él, ami mentén egy csúcs elértté vált.

uv **előreél:** nem faél, de u -ból v -be faélekből irányított út vezet.

uv **visszaél:** v -ből u -ba faélekből irányított út vezet.

keresztél: minden más él (u és v közt nincs leszármazási viszony).

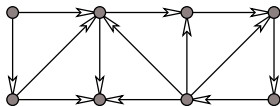
(3) A **bejárás fája:** a faélek alkotta részgráf.

Megf: Irányítatlan esetben az előreél és a visszaél ugyanazt jelenti.

Terminológia: Ha a bejárás fájában u -ból v -be irányított út vezet, akkor u a v **őse** és v az u **leszármazottja**. A faél és az előreél tehát ősből leszármazottba, a visszaél pedig leszármazottból ősbé vezet.

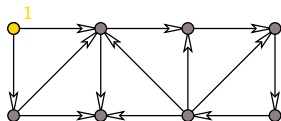
¹A gyökér már kezdetben **elért** állapotú, így kivétel az általános szabály alól.

A BFS és tulajdonságai



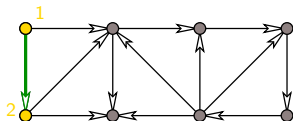
Szélességi bejárás (BFS) szabálya: Olyan bejárás, ahol az 1. esetben mindig a legkorábban elért u csúcsot választjuk. Nézzük meg egy **irányított** gráf BFS bejárását.

A BFS és tulajdonságai



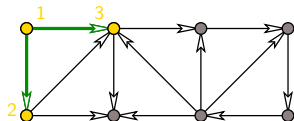
Szélességi bejárás (BFS) szabálya: Olyan bejárás, ahol az 1. esetben mindig a legkorábban elért u csúcsot választjuk. Nézzük meg egy **irányított** gráf BFS bejárását.

A BFS és tulajdonságai



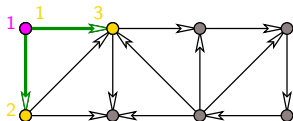
Szélességi bejárás (BFS) szabálya: Olyan bejárás, ahol az 1. esetben mindig a legkorábban elért u csúcsot választjuk. Nézzük meg egy **irányított** gráf BFS bejárását.

A BFS és tulajdonságai



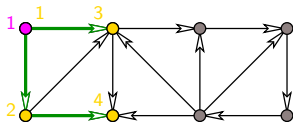
Szélességi bejárás (BFS) szabálya: Olyan bejárás, ahol az 1. esetben mindig a legkorábban elért u csúcsot választjuk. Nézzük meg egy **irányított** gráf BFS bejárását.

A BFS és tulajdonságai



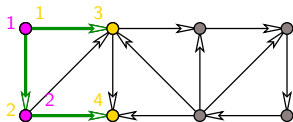
Szélességi bejárás (BFS) szabálya: Olyan bejárás, ahol az 1. esetben mindig a legkorábban elért u csúcsot választjuk. Nézzük meg egy **irányított** gráf BFS bejárását.

A BFS és tulajdonságai



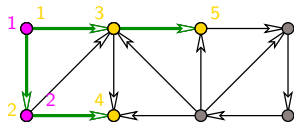
Szélességi bejárás (BFS) szabálya: Olyan bejárás, ahol az 1. esetben mindig a legkorábban elért u csúcsot választjuk. Nézzük meg egy **irányított** gráf BFS bejárását.

A BFS és tulajdonságai



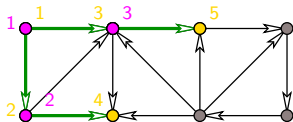
Szélességi bejárás (BFS) szabálya: Olyan bejárás, ahol az 1. esetben mindig a legkorábban elért u csúcsot választjuk. Nézzük meg egy **irányított** gráf BFS bejárását.

A BFS és tulajdonságai



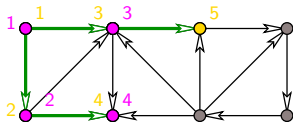
Szélességi bejárás (BFS) szabálya: Olyan bejárás, ahol az 1. esetben mindig a legkorábban elért u csúcsot választjuk. Nézzük meg egy **irányított** gráf BFS bejárását.

A BFS és tulajdonságai



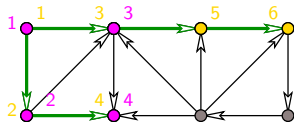
Szélességi bejárás (BFS) szabálya: Olyan bejárás, ahol az 1. esetben mindig a legkorábban elért u csúcsot választjuk. Nézzük meg egy **irányított** gráf BFS bejárását.

A BFS és tulajdonságai



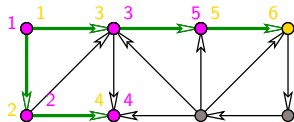
Szélességi bejárás (BFS) szabálya: Olyan bejárás, ahol az 1. esetben mindig a legkorábban elért u csúcsot választjuk. Nézzük meg egy **irányított** gráf BFS bejárását.

A BFS és tulajdonságai



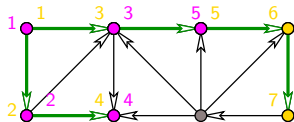
Szélességi bejárás (BFS) szabálya: Olyan bejárás, ahol az 1. esetben mindig a legkorábban elért u csúcsot választjuk. Nézzük meg egy **irányított** gráf BFS bejárását.

A BFS és tulajdonságai



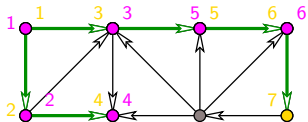
Szélességi bejárás (BFS) szabálya: Olyan bejárás, ahol az 1. esetben mindig a legkorábban elért u csúcsot választjuk. Nézzük meg egy **irányított** gráf BFS bejárását.

A BFS és tulajdonságai



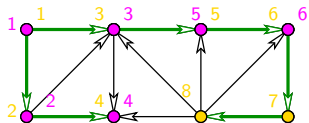
Szélességi bejárás (BFS) szabálya: Olyan bejárás, ahol az 1. esetben mindig a legkorábban elért u csúcsot választjuk. Nézzük meg egy **irányított** gráf BFS bejárását.

A BFS és tulajdonságai



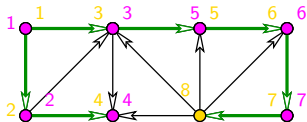
Szélességi bejárás (BFS) szabálya: Olyan bejárás, ahol az 1. esetben mindig a legkorábban elért u csúcsot választjuk. Nézzük meg egy **irányított** gráf BFS bejárását.

A BFS és tulajdonságai



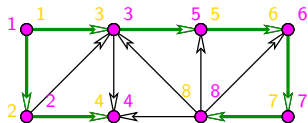
Szélességi bejárás (BFS) szabálya: Olyan bejárás, ahol az 1. esetben mindig a legkorábban elért u csúcsot választjuk. Nézzük meg egy **irányított** gráf BFS bejárását.

A BFS és tulajdonságai



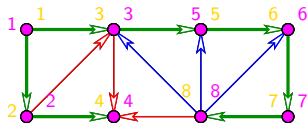
Szélességi bejárás (BFS) szabálya: Olyan bejárás, ahol az 1. esetben mindig a legkorábban elért u csúcsot választjuk. Nézzük meg egy **irányított** gráf BFS bejárását.

A BFS és tulajdonságai



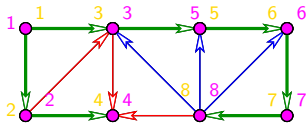
Szélességi bejárás (BFS) szabálya: Olyan bejárás, ahol az 1. esetben mindig a legkorábban elért u csúcsot választjuk. Nézzük meg egy **irányított** gráf BFS bejárását.

A BFS és tulajdonságai

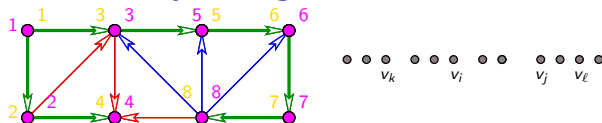


Szélességi bejárás (BFS) szabálya: Olyan bejárás, ahol az 1. esetben mindig a legkorábban elért u csúcsot választjuk. Nézzük meg egy **irányított** gráf BFS bejárását.

A BFS és tulajdonságai



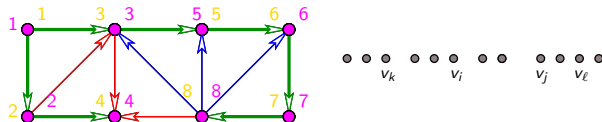
A BFS és tulajdonságai



Állítás: Tfh $G = (V, E)$ **BFS bejárása után** a csúcsok elérési sorrendje $v_1, v_2, \dots, v_n$. Ekkor az alábbiak teljesülnek.

(1) Ha $i < j$, akkor v_i -t hamarabb fejezzük be, mint v_j -t, továbbá v_i gyerekei az elérési sorrendben megelőzik v_j gyerekeit.

A BFS és tulajdonságai

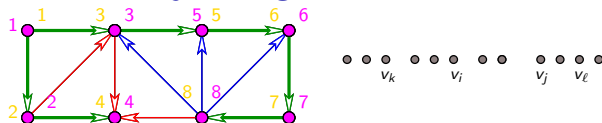


Állítás: Tfh $G = (V, E)$ **BFS bejárása után** a csúcsok elérési sorrendje $v_1, v_2, \dots, v_n$. Ekkor az alábbiak teljesülnek.

(1) Ha $i < j$, akkor v_i -t hamarabb fejezzük be, mint v_j -t, továbbá v_i gyerekei az elérési sorrendben megelőzik v_j gyerekeit.

Biz: A v_i befejezésének pillanatában v_i minden gyereke elért, de v_j -nek még egy gyereke sem az. Ezért v_j gyerekeit a v_i csúcs befejezése után érjük el, majd ezt követően fejezzük be v_j -t. □

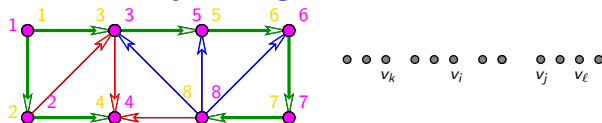
A BFS és tulajdonságai



Állítás: Tfh $G = (V, E)$ **BFS bejárása után** a csúcsok elérési sorrendje $v_1, v_2, \dots, v_n$. Ekkor az alábbiak teljesülnek.

(1) Ha $i < j$, akkor v_i -t hamarabb fejezzük be, mint v_j -t, továbbá v_i gyerekei az elérési sorrendben megelőzik v_j gyerekeit.

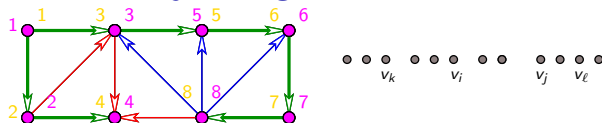
A BFS és tulajdonságai



Állítás: Tfh $G = (V, E)$ **BFS bejárása után** a csúcsok elérési sorrendje $v_1, v_2, \dots, v_n$. Ekkor az alábbiak teljesülnek.

- (1) Ha $i < j$, akkor v_i -t hamarabb fejezzük be, mint v_j -t, továbbá v_i gyerekei az elérési sorrendben megelőzik v_j gyerekeit.
- (2) **Az elérési és befejezési sorrend megegyezik.**

A BFS és tulajdonságai



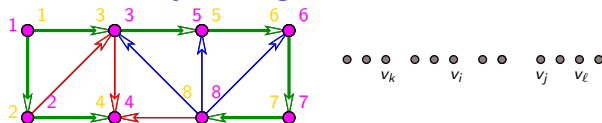
Állítás: Tfh $G = (V, E)$ **BFS bejárása után** a csúcsok elérési sorrendje $v_1, v_2, \dots, v_n$. Ekkor az alábbiak teljesülnek.

(1) Ha $i < j$, akkor v_i -t hamarabb fejezzük be, mint v_j -t, továbbá v_i gyerekei az elérési sorrendben megelőzik v_j gyerekeit.

(2) **Az elérési és befejezési sorrend megegyezik.**

Biz: Ha v_i -t korábban érjük el, mint v_j -t, akkor (1) miatt v_i -t korábban is fejezzük be v_j -nél. Ezért bármely két csúcs sorrendje ugyanaz az elérési sorrendben mint befejezési sorrendben. Tehát az elérési sorrendnek meg kell egyeznie a befejezési sorrenddel. $\square$

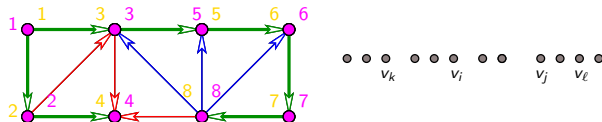
A BFS és tulajdonságai



Állítás: Tfh $G = (V, E)$ **BFS bejárása után** a csúcsok elérési sorrendje $v_1, v_2, \dots, v_n$. Ekkor az alábbiak teljesülnek.

- (1) Ha $i < j$, akkor v_i -t hamarabb fejezzük be, mint v_j -t, továbbá v_i gyerekei az elérési sorrendben megelőzik v_j gyerekeit.
- (2) **Az elérési és befejezési sorrend megegyezik.**

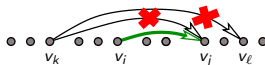
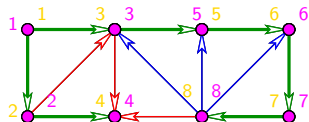
A BFS és tulajdonságai



Állítás: Tfh $G = (V, E)$ **BFS bejárása után** a csúcsok elérési sorrendje $v_1, v_2, \dots, v_n$. Ekkor az alábbiak teljesülnek.

- (1) Ha $i < j$, akkor v_i -t hamarabb fejezzük be, mint v_j -t, továbbá v_i gyerekei az elérési sorrendben megelőzik v_j gyerekeit.
- (2) **Az elérési és befejezési sorrend megegyezik.**
- (3) **Gráfél nem ugorhat át faélt:** ha $k < i < j \leq \ell$ és $v_i v_j$ faél, akkor $v_k v_\ell$ nem lehet gráfél.

A BFS és tulajdonságai



Állítás: Tfh $G = (V, E)$ **BFS bejárása után** a csúcsok elérési sorrendje $v_1, v_2, \dots, v_n$. Ekkor az alábbiak teljesülnek.

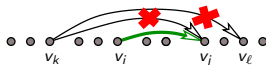
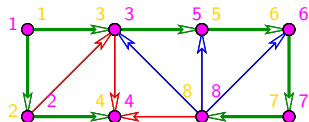
(1) Ha $i < j$, akkor v_i -t hamarabb fejezzük be, mint v_j -t, továbbá v_i gyerekei az elérési sorrendben megelőzik v_j gyerekeit.

(2) **Az elérési és befejezési sorrend megegyezik.**

(3) **Gráfél nem ugorhat át faélt:** ha $k < i < j \leq \ell$ és $v_i v_j$ faél, akkor $v_k v_\ell$ nem lehet gráfél.

Biz: Indirekt bizonyítunk. Ha $v_k v_\ell \in E(G)$, akkor v_ℓ szülője v_k vagy egy v_k -t megelőző csúcs. (1) miatt v_j szülője sem következhet v_k után, vagyis v_i nem lehet v_j szülője. Ellentmondás. $\square$

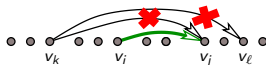
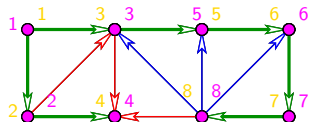
A BFS és tulajdonságai



Állítás: Tfh $G = (V, E)$ **BFS bejárása után** a csúcsok elérési sorrendje $v_1, v_2, \dots, v_n$. Ekkor az alábbiak teljesülnek.

- (1) Ha $i < j$, akkor v_i -t hamarabb fejezzük be, mint v_j -t, továbbá v_i gyerekei az elérési sorrendben megelőzik v_j gyerekeit.
- (2) **Az elérési és befejezési sorrend megegyezik.**
- (3) **Gráfél nem ugorhat át faélt:** ha $k < i < j \leq \ell$ és $v_i v_j$ faél, akkor $v_k v_\ell$ nem lehet gráfél.

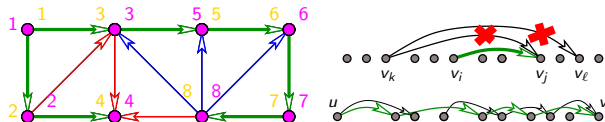
A BFS és tulajdonságai



Állítás: Tfh $G = (V, E)$ **BFS bejárása után** a csúcsok elérési sorrendje $v_1, v_2, \dots, v_n$. Ekkor az alábbiak teljesülnek.

- (1) Ha $i < j$, akkor v_i -t hamarabb fejezzük be, mint v_j -t, továbbá v_i gyerekei az elérési sorrendben megelőzik v_j gyerekeit.
- (2) **Az elérési és befejezési sorrend megegyezik.**
- (3) **Gráfél nem ugorhat át faélt:** ha $k < i < j \leq \ell$ és $v_i v_j$ faél, akkor $v_k v_\ell$ nem lehet gráfél.
- (4) Ha P a BFS-fa uv -útja, akkor P egy legrövidebb uv -út G -ben.

A BFS és tulajdonságai



Állítás: Tfh $G = (V, E)$ **BFS bejárása után** a csúcsok elérési sorrendje $v_1, v_2, \dots, v_n$. Ekkor az alábbiak teljesülnek.

(1) Ha $i < j$, akkor v_i -t hamarabb fejezzük be, mint v_j -t, továbbá v_i gyerekei az elérési sorrendben megelőzik v_j gyerekeit.

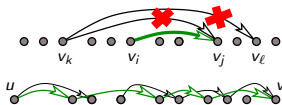
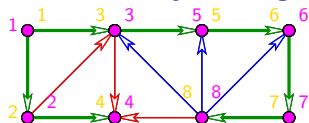
(2) **Az elérési és befejezési sorrend megegyezik.**

(3) **Gráfél nem ugorhat át faélt:** ha $k < i < j \leq \ell$ és $v_i v_j$ faél, akkor $v_k v_\ell$ nem lehet gráfél.

(4) Ha P a BFS-fa uv -útja, akkor P egy legrövidebb uv -út G -ben.

Biz: Ha P' egy G -beli uv -út, akkor P' egyetlen éle sem ugorhat át P -beli élt. Ezért P' utolsó éle nem kezdődhet korábban P utolsó élénél. Hasonló igaz P' utolsó előtti, stb. éleire. Így v -ből nem lehet u -ba visszajutni P élszámánál kevesebb élen. $\square$

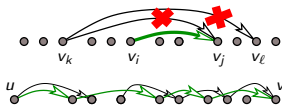
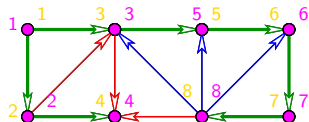
A BFS és tulajdonságai



Állítás: Tfh $G = (V, E)$ **BFS bejárása után** a csúcsok elérési sorrendje $v_1, v_2, \dots, v_n$. Ekkor az alábbiak teljesülnek.

- (1) Ha $i < j$, akkor v_i -t hamarabb fejezzük be, mint v_j -t, továbbá v_i gyerekei az elérési sorrendben megelőzik v_j gyerekeit.
- (2) **Az elérési és befejezési sorrend megegyezik.**
- (3) **Gráfél nem ugorhat át faél:** ha $k < i < j \leq \ell$ és $v_i v_j$ faél, akkor $v_k v_\ell$ nem lehet gráfél.
- (4) Ha P a BFS-fa uv -útja, akkor P egy legrövidebb uv -út G -ben.

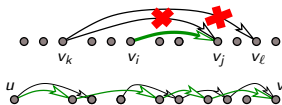
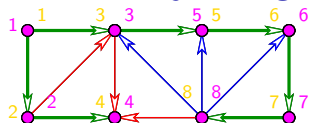
A BFS és tulajdonságai



Állítás: Tfh $G = (V, E)$ **BFS bejárása után** a csúcsok elérési sorrendje $v_1, v_2, \dots, v_n$. Ekkor az alábbiak teljesülnek.

- (1) Ha $i < j$, akkor v_i -t hamarabb fejezzük be, mint v_j -t, továbbá v_i gyerekei az elérési sorrendben megelőzik v_j gyerekeit.
- (2) **Az elérési és befejezési sorrend megegyezik.**
- (3) **Gráfél nem ugorhat át faél:** ha $k < i < j \leq \ell$ és $v_i v_j$ faél, akkor $v_k v_\ell$ nem lehet gráfél.
- (4) Ha P a BFS-fa uv -útja, akkor P egy legrövidebb uv -út G -ben.
- (5) **A BFS-fa egy legrövidebb utak fája:** a BFS-fa v_1 gyökeréből mely v_i csúcsba vezető faút a G egy legkevesebb élű $v_1 v_i$ -útja.

A BFS és tulajdonságai

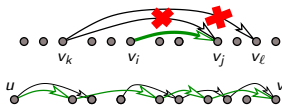
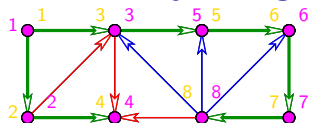


Állítás: Tfh $G = (V, E)$ **BFS bejárása után** a csúcsok elérési sorrendje $v_1, v_2, \dots, v_n$. Ekkor az alábbiak teljesülnek.

- (1) Ha $i < j$, akkor v_i -t hamarabb fejezzük be, mint v_j -t, továbbá v_i gyerekei az elérési sorrendben megelőzik v_j gyerekeit.
- (2) **Az elérési és befejezési sorrend megegyezik.**
- (3) **Gráfél nem ugorhat át faél:** ha $k < i < j \leq \ell$ és $v_i v_j$ faél, akkor $v_k v_\ell$ nem lehet gráfél.
- (4) Ha P a BFS-fa uv -útja, akkor P egy legrövidebb uv -út G -ben.
- (5) **A BFS-fa egy legrövidebb utak fája:** a BFS-fa v_1 gyökeréből mely v_i csúcsba vezető faút a G egy legkevesebb élű $v_1 v_i$ -útja.

Biz: Közvetlenül adódik (4)-ből.

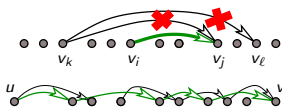
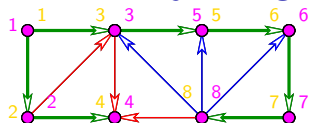
A BFS és tulajdonságai



Állítás: Tfh $G = (V, E)$ **BFS bejárása után** a csúcsok elérési sorrendje $v_1, v_2, \dots, v_n$. Ekkor az alábbiak teljesülnek.

- (1) Ha $i < j$, akkor v_i -t hamarabb fejezzük be, mint v_j -t, továbbá v_i gyerekei az elérési sorrendben megelőzik v_j gyerekeit.
- (2) **Az elérési és befejezési sorrend megegyezik.**
- (3) **Gráfél nem ugorhat át faél:** ha $k < i < j \leq \ell$ és $v_i v_j$ faél, akkor $v_k v_\ell$ nem lehet gráfél.
- (4) Ha P a BFS-fa uv -útja, akkor P egy legrövidebb uv -út G -ben.
- (5) **A BFS-fa egy legrövidebb utak fája:** a BFS-fa v_1 gyökeréből mely v_i csúcsba vezető faút a G egy legkevesebb élű $v_1 v_i$ -útja.

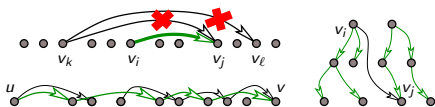
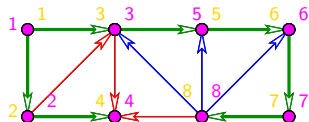
A BFS és tulajdonságai



Állítás: Tfh $G = (V, E)$ **BFS bejárása után** a csúcsok elérési sorrendje $v_1, v_2, \dots, v_n$. Ekkor az alábbiak teljesülnek.

- (1) Ha $i < j$, akkor v_i -t hamarabb fejezzük be, mint v_j -t, továbbá v_i gyerekei az elérési sorrendben megelőzik v_j gyerekeit.
- (2) **Az elérési és befejezési sorrend megegyezik.**
- (3) **Gráfél nem ugorhat át faélt:** ha $k < i < j \leq \ell$ és $v_i v_j$ faél, akkor $v_k v_\ell$ nem lehet gráfél.
- (4) Ha P a BFS-fa uv -útja, akkor P egy legrövidebb uv -út G -ben.
- (5) **A BFS-fa egy legrövidebb utak fája:** a BFS-fa v_1 gyökeréből bármely v_i csúcsba vezető faút a G egy legkevesebb élű $v_1 v_i$ -útja.
- (6) Minden él legfeljebb egy szintet lép lefelé a BFS-fában, így **nincs előreél.** (Írányítatlan esetben csak faél és keresztél van.)

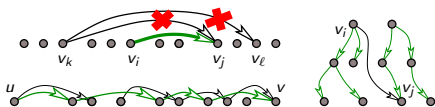
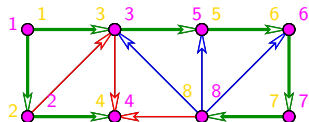
A BFS és tulajdonságai



Állítás: Tfh $G = (V, E)$ **BFS bejárása után** a csúcsok elérési sorrendje $v_1, v_2, \dots, v_n$. Ekkor az alábbiak teljesülnek.

- (1) Ha $i < j$, akkor v_i -t hamarabb fejezzük be, mint v_j -t, továbbá v_i gyerekei az elérési sorrendben megelőzik v_j gyerekeit.
 - (2) **Az elérési és befejezési sorrend megegyezik.**
 - (3) **Gráfél nem ugorhat át faélt:** ha $k < i < j \leq \ell$ és $v_i v_j$ faél, akkor $v_k v_\ell$ nem lehet gráfél.
 - (4) Ha P a BFS-fa uv -útja, akkor P egy legrövidebb uv -út G -ben.
 - (5) **A BFS-fa egy legrövidebb utak fája:** a BFS-fa v_1 gyökeréből mely v_i csúcsba vezető faút a G egy legkevesebb élű $v_1 v_i$ -útja.
 - (6) Minden él legfeljebb egy szintet lép lefelé a BFS-fában, így **nincs előreél**. (Írányítatlan esetben csak faél és keresztél van.)
- Biz:** Ha $v_i v_j$ legalább két szintet lép lefelé, akkor lenne rövidebb út v_1 -ből v_j -be a BFS-fabelinél. Ez ellentmond (5)-nek.

A BFS és tulajdonságai

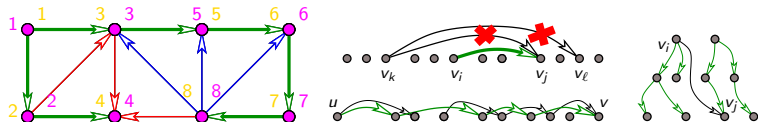


Állítás: Tfh $G = (V, E)$ **BFS bejárása után** a csúcsok elérési sorrendje $v_1, v_2, \dots, v_n$. Ekkor az alábbiak teljesülnek.

- (1) Ha $i < j$, akkor v_i -t hamarabb fejezzük be, mint v_j -t, továbbá v_i gyerekei az elérési sorrendben megelőzik v_j gyerekeit.
- (2) **Az elérési és befejezési sorrend megegyezik.**
- (3) **Gráfél nem ugorhat át faélt:** ha $k < i < j \leq \ell$ és $v_i v_j$ faél, akkor $v_k v_\ell$ nem lehet gráfél.
- (4) Ha P a BFS-fa uv -útja, akkor P egy legrövidebb uv -út G -ben.
- (5) **A BFS-fa egy legrövidebb utak fája:** a BFS-fa v_1 gyökeréből mely v_i csúcsba vezető faút a G egy legkevesebb élű $v_1 v_i$ -útja.
- (6) Minden él legfeljebb egy szintet lép lefelé a BFS-fában, így **nincs előreél.** (Írányítatlan esetben csak faél és keresztél van.)

Biz:

A BFS és tulajdonságai



Állítás: Tfh $G = (V, E)$ **BFS bejárása után** a csúcsok elérési sorrendje $v_1, v_2, \dots, v_n$. Ekkor az alábbiak teljesülnek.

- (1) Ha $i < j$, akkor v_i -t hamarabb fejezzük be, mint v_j -t, továbbá v_i gyerekei az elérési sorrendben megelőzik v_j gyerekeit.
 - (2) **Az elérési és befejezési sorrend megegyezik.**
 - (3) **Gráfél nem ugorhat át faélt:** ha $k < i < j \leq \ell$ és $v_i v_j$ faél, akkor $v_k v_\ell$ nem lehet gráfél.
 - (4) Ha P a BFS-fa uv -útja, akkor P egy legrövidebb uv -út G -ben.
 - (5) **A BFS-fa egy legrövidebb utak fája:** a BFS-fa v_1 gyökeréből bármely v_i csúcsba vezető faút a G egy legkevesebb élű $v_1 v_i$ -útja.
 - (6) Minden él legfeljebb egy szintet lép lefelé a BFS-fában, így **nincs előreél**. (Írányítatlan esetben csak faél és keresztél van.)
- Biz:** Ha pedig lenne előreél, akkor az legalább két szintet lépne lefelé, amiről most láttuk, hogy lehetetlen.



Mit tanultunk ma?

Mit tanultunk ma?

- ▶ Gráfbejárás fogalma
(csúcsok evolúciója, bejárési sorrendek, élek osztályozása, bejárás fája)

Mit tanultunk ma?

- ▶ Gráfbejárás fogalma
(csúcsok evolúciója, bejárési sorrendek, élek osztályozása, bejárás fája)
- ▶ BFS bejárás

Mit tanultunk ma?

- ▶ Gráfbejárás fogalma
(csúcsok evolúciója, bejárési sorrendek, élek osztályozása, bejárás fája)
- ▶ BFS bejárás
- ▶ BFS során az elérési és befejezési sorrend megegyezik, nincs se faélt átugró gráfél, se előreél, a BFS-fa pedig egy legrövidebb utak fája

Mit tanultunk ma?

- ▶ Gráfbejárás fogalma
(csúcsok evolúciója, bejárési sorrendek, élek osztályozása, bejárás fája)
- ▶ BFS bejárás
- ▶ BFS során az elérési és befejezési sorrend megegyezik, nincs se faélt átugró gráfél, se előreél, a BFS-fa pedig egy legrövidebb utak fája

Köszönöm a figyelmet!