

Algoritmuselmélet 11. előadás

Katona Gyula Y.
Budapesti Műszaki és Gazdaságtudományi Egyetem
Számítástudományi Tsz.
I. B. 137/b
kiskat@cs.bme.hu

2002 Március 26.

Kruskal módszere

Mindig a legkisebb súlyú olyan élet színezzük **kék**re, ami még nem alkot kört az eddigi **kék** élekkel.

⇒ A **kék** élek végig egy erdőt határoznak meg, akkor van kész, ha feszítőfa lesz.

⇒ Az új **kék** él az eddigi erdő két komponensét fogja összekötni. Kezdetben n komponens, egy él színezésével eggyel csökken a komponensek száma.

procedure Kruskal (G : gráf; **var** F, H : élek halmaza);
begin

$F := \emptyset$; $H := E$;

while $H \neq \emptyset$ **do begin**

Töröljük a H minimális súlyú (v, w) élet.

Ha $F \cup \{(v, w)\}$ -ben nincs kör

(azaz a v, w pontok különböző **kék** fákbán vannak), akkor

$F := F \cup \{(v, w)\}$

end

end
⇒ Ha a (v, w) él kört eredményez, akkor **piros** él, ha pedig nem, akkor **kék** éle.

Tétel. A Kruskal-algoritmus eredményeként végül F a G gráf egy minimális költségű feszítőfájának éleit tartalmazza.

Bizonyítás: ez is piros-kék algoritmus

JAVA animáció: Kruskal

Implementáció:

- élekből kupacot építünk $\rightarrow O(e \log e)$ lépés
- éleket előre rendezzük $\rightarrow O(e \log e)$ lépés

Hogyan döntjük el, hogy a kiválasztott él kört alkot-e az eddigi kiválasztottakkal?

\implies Tartsuk nyilván az aktuálisan egy kék fába tartozó csúcsokat mint halmazokat.

Kell: UNIÓ, HOLVAN

Az UNIÓ-HOLVAN adatszerkezet

Legyen adott egy véges S halmaz. Ennek egy partícióját szeretnénk tárolni

$$\rightarrow U_1, \dots, U_k \subseteq S$$

Adott egy n elemű S halmaz, és ennek bizonyos U_1, \dots, U_m részhalmazai, melyekre $U_i \cap U_j = \emptyset$ ($i \neq j$) és $U_1 \cup \dots \cup U_m = S$ (vagyis az U_j részhalmazok S egy partícióját adják).

Műveletek:

UNIÓ(U_i, U_j) = $(\{U_1, \dots, U_m\} \cup \{U_i \cup U_j\}) \setminus \{U_i, U_j\}$ (az U_i, U_j halmazokat $U_i \cup U_j$ helyettesíti).

HOLVAN(v) eredménye (itt $v \in S$) annak az U_i halmaznak a neve, amelynek v eleme.

Kruskalban:

annak megvizsgálása, hogy milyen színű legyen az új (u, v) él

\implies Ha **HOLVAN**(u) \neq **HOLVAN**(v), akkor **kék**, különben **piros**.

Új él színezése \implies **UNIÓ**(**HOLVAN**(u), **HOLVAN**(v))

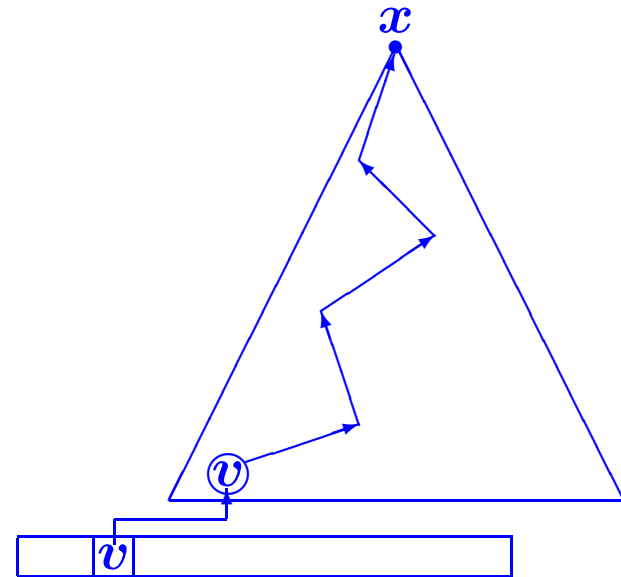
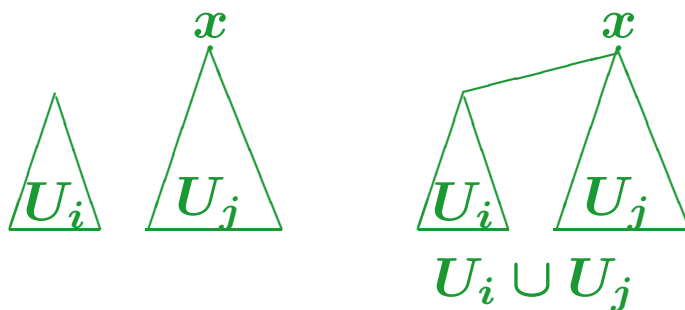
Implementáció fákkal

$U_j \rightarrow$ gyökeres, felfelé irányított fa

U_j elemeit a fa csúcsaiban tároljuk, egy szülőmutatóval.

Egy részhalmaz *neve* legyen az őt ábrázoló fa gyökere. A gyökérben nyilvántartjuk még a fa méretét is.

- **UNIÓ:** $U_i \cup U_j$ fáját a következőképpen készítjük el:
Tegyük fel, hogy $|U_i| \leq |U_j|$. Ekkor az U_j fa x gyökeréhez gyermekként hozzákapcsoljuk U_i gyökerét.



- **HOLVAN:** A $v \in S$ elemet tartalmazó részhalmaz nevét, azaz a megfelelő fa gyökerét a szülőkhöz menő mutatók végigkövetésével találhatjuk meg.

Az UNIÓ hívásakor az U_i és U_j halmazok a gyökérükkel adottak

⇒ költség: $O(1)$

HA egy v csúcs új gyökér alá kerül, akkor egy szinttel lesz távolabb a gyökértől, míg az új fájának a mérete legalább az eredeti duplájára változik.

⇒ Egy csúcs legfeljebb $\log_2 n$ -szer kerülhet új gyökér alá

⇒ szintszám $\leq \log_2 n$

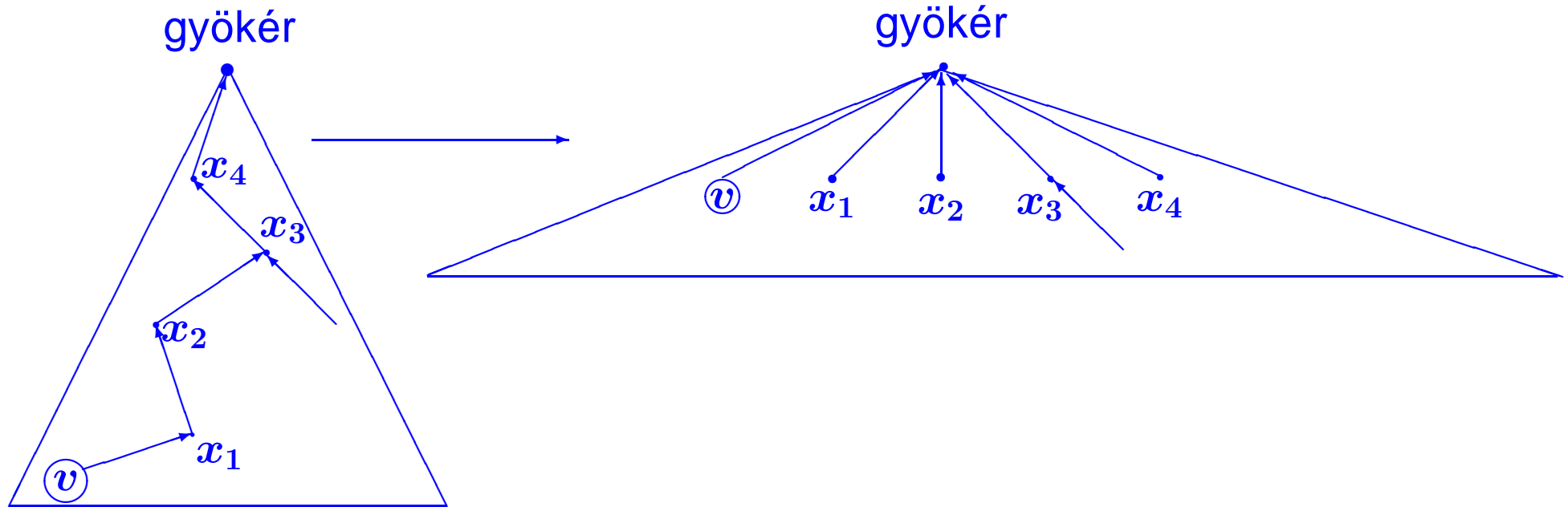
⇒ HOLVAN költsége $O(\log n)$

Tétel. A Kruskal-algoritmus költsége $O(e \log e)$. □

Bizonyítás: $2e$ HOLVAN, és $n - 1$ UNIÓ műveletet jelent. Ezek időigénye $O(e \log n + n) = O(e \log n)$, vagy ami ugyanaz: $O(e \log e)$.

A HOLVAN gyorsítása: útösszenyomás

Egy pontról többször is megnézzük HOLVAN, mindig $\log n$ lépés
 \implies Az első alkalommal, minden őst kössük közvetlenül a gyökérbe



Tétel. Legyen $|S| = n$, és tegyük fel, hogy kezdetben mindegyik U_j egyelemű. Ha egy olyan utasítássorozatot hajtunk végre (útösszenyomással), melyben $n - 1$ UNIÓ és $m \geq n - 1$ HOLVAN szerepel, akkor ennek az időigénye $O(m\alpha(m))$.

A korlátban szereplő $\alpha : \mathbb{Z}^+ \rightarrow \mathbb{Z}^+$ függvény az *inverz Ackermann-függvény*.

$\alpha(m)$ a végtelenhez tart, ha $m \rightarrow \infty$, de nagyon lassan, lassabban mint a logaritmus k -szori önmagába helyettesítésével adódó $\log \log \cdots \log m$ függvény ($k \in \mathbb{Z}^+$ tetszőleges).

Pl. $\alpha(m) \leq 4$, ha $m < 2^{65536}$

A normális méretű feladatoknál tehát $\alpha(m)$ állandónak (≤ 4) tekinthető.

Tétel. *Ha az élek rendezésével kapcsolatos teendők $O(e)$ időben megoldhatók, akkor a Kruskal-algoritmus $O(e\alpha(e))$ időben megvalósítható.*

Manipuláció a súlyokkal \implies Yao (1975): $O(e \log \log n)$

Véletlen módszerek \implies D. R. Karger, P. N. Klein, R. E. Tarjan, (1994):
várhatóan $O(e)$

On-line változatban is működik a **piros-kék** algoritmus: színezzük az új éleket kékre; ha emiatt kialakul egy kék kör, akkor abból töröljük egy maximális súlyú éleket.

JAVA animáció: Kruskal

Maximális párosítás páros gráfokban

Definíció. A $G = (V, E)$ gráfot **párosnak** nevezzük, ha V csúcshalmaza felosztható két diszjunkt részre – V_1 -re és V_2 -re – úgy, hogy minden él ezen két halmaz között fut, vagyis $(x, y) \in E$ esetén $x \in V_1$ és $y \in V_2$ vagy fordítva.

Definíció. Legyen $G = (V, E)$ egy tetszőleges gráf. Az E élhalmaz $E' \subseteq E$ részhalmaza G egy **párosítása**, ha a $G' = (V, E')$ gráfban minden pont foka legfeljebb egy.

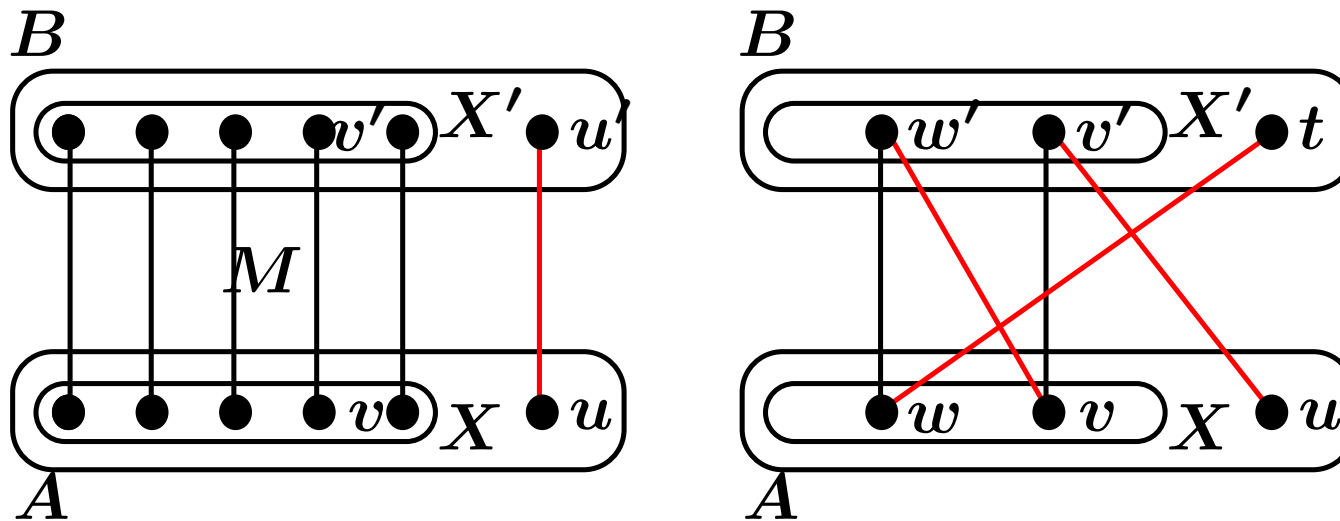
Definíció. A G gráf egy E' párosítása **maximális**, ha G minden E'' párosítására $|E''| \leq |E'|$.

A probléma: Adott egy $G = (V_1, V_2; E)$ páros gráf. Határozzuk meg G egy maximális párosítását.

Definíció. Legyen G egy tetszőleges gráf, és E' a G egy párosítása. Egy G -beli utat **E' -alternáló útnak** hívunk, ha felváltva tartalmaz párosított és nem párosított éleket.

Definíció. Legyen E' a $G = (V, E)$ gráf egy párosítása. Ekkor egy olyan E' -alternáló út, melynek mindkét végpontja párosítatlan, **E' -re nézve javító út**, vagy röviden **E' -javító út**.

Tétel. Legyen $G = (V, E)$ egy tetszőleges gráf és E' egy párosítása. Ha E' -re nézve nincs javító út G -ben, akkor E' a G egy maximális párosítása.



Hogyan keressünk javító utat?

Javító út keresése alternáló erdő építésével

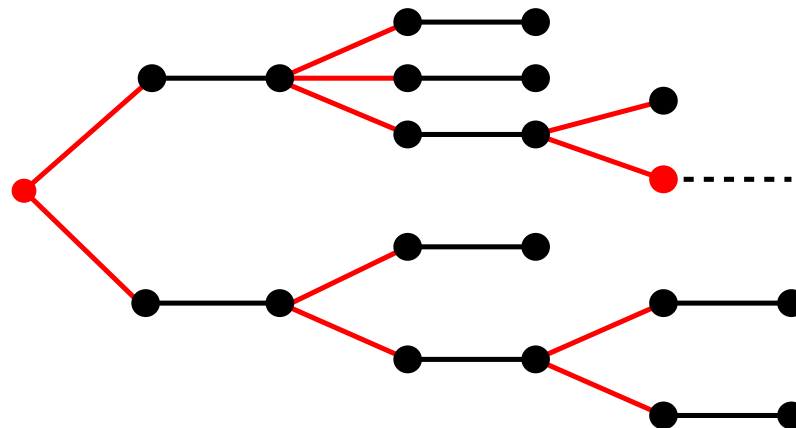
0. szint: V_1 azon pontjai, melyeket E' nem fed le, vagyis a párosítatlan pontok.

⋮

$2k - 1$. szint: V_2 azon még fel nem vett pontjai, melyek egy párosítatlan, azaz egy $E \setminus E'$ -beli éllel elérhetők egy $2k - 2$. szintbeli pontból; ezen éllel együtt.

$2k$. szint: V_1 azon még fel nem vett pontjai, melyek egy párosított, azaz egy E' -beli éllel elérhetők egy $2k - 1$. szintbeli pontból; ezen éllel együtt.

⋮



Tétel. *A $G = (V_1, V_2; E)$ páros gráfban akkor és csak akkor van az E' párosításra nézve javító út, ha az E' -hez tartozó alternáló erdőben valamelyik páratlan szinten megjelenik egy párosítatlan pont.*

Bizonyítás: Ha találtunk páratlan szinten párosítatlan utat, akkor a gyökér felé vezető út javító út ✓

Fordítva:

Megmutatjuk, hogy a V_1 párosítatlan csúcsaiból (ezek vannak az erdőben a nulladik szinten) alternáló úton elérhető pontok mindegyikét beválasztjuk valamikor az alternáló erdőbe.

Tegyük fel, hogy v_0, v_1, \dots, v_k egy alternáló út, és $v_0 \in V_1$ egy párosítatlan csúcs; i szerinti indukcióval megmutatjuk, hogy v_i bekerül az erdőbe.

Ha $i = 0$ ✓

Az út v_{2j} csúcsa V_1 -ben van és a (v_{2j}, v_{2j+1}) él párosítatlan. \implies ha v_{2j} -t beválasztottuk, akkor v_{2j+1} is bekerül

Az út v_{2j-1} csúcsa V_2 -ben van és $(v_{2j-1}, v_{2j}) \in E'$. Így v_{2j-1} után v_{2j} is sorra kerül, ha korábban ez még nem történt meg.

Tegyük fel, hogy G -ben a v és w csúcsok egy javító út végpontjai.

$v \in V_1$ párosítatlan $\implies w \in V_2$

w elérhető alternáló úton v -ből \implies valamikor beválasztjuk

De V_2 -beli csúcsokat csak páratlan szintekre veszünk fel $\implies w$ is itt lesz

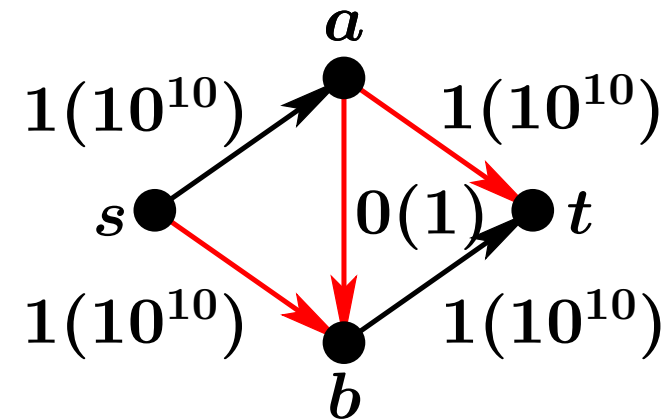
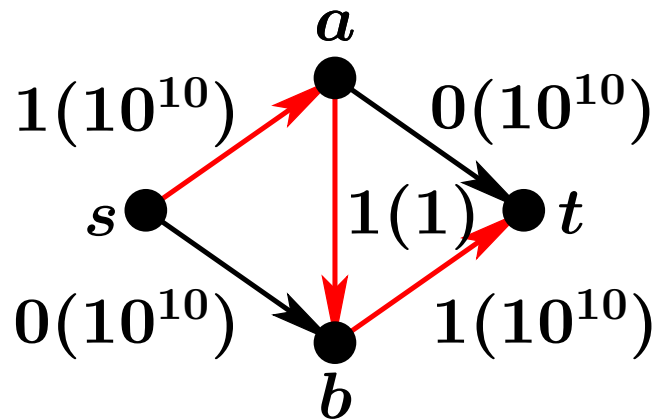
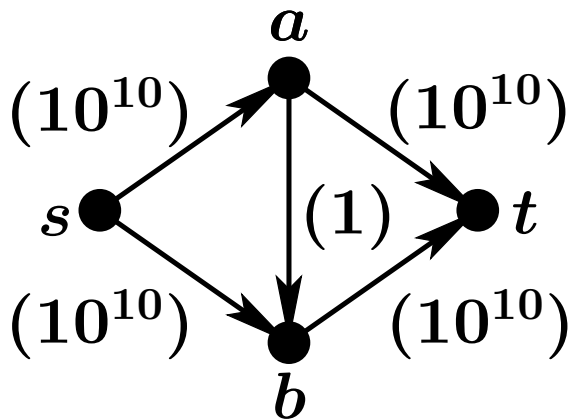
Lépésszám: Alternáló erdő építése: $O(e)$, össze lépésszám: $O(ne)$

Karp (1973): $O(e\sqrt{n})$

Maximális folyamok hálózatokban

JAVA animáció: Ford-Fulkerson algoritmus

Ha minden kapacitás egész és a maximális folyam értéke f , akkor legfeljebb f javítással megkapjuk a maximális folyamot.



Ha az élkapacitások racionális számok \implies véges lépés

Ha irracionális kapacitásokat is megengedünk \implies lehet, hogy nem ér véget véges sok lépésben, sőt lehet, hogy nem is jó értékhez konvergál

Edmonds–Karp algoritmus

A folyam növelésére mindig a legrövidebb – vagyis a legkevesebb élből álló – növelő utak egyikét válasszuk.

Tekintsük a G_f javító gráfot; legyen benne π egy legrövidebb növelő út. Ennek hosszát (élszámát) jelölje l .

Szélességi kereséssel osszuk szintekre $\implies D[v]$

(1) Egy él legfeljebb egy réteggel mehet előre.

A G_f egy $x \rightarrow y$ élét nevezzük *vastagnak*, ha $D[y] = D[x] + 1$.

(2) Az l hosszúságú $s \rightsquigarrow t$ utak csupa vastag élből állnak, és nincs l -nél rövidebb $s \rightsquigarrow t$ út.

\Leftarrow Legrövidebb útnak muszáj mindig feljebb menni

Mi történik, ha javítunk π mentén?

Legalább egy él telítődik és eltűnik a javító gráfból.

Legfeljebb l darab új él jelenik meg a $G_{f'}$ -ben (π élei ellenkező irányítással, ha eddig még 0 folyt át rajtuk).

\implies *a következő növelő út sem lehet rövidebb l -nél.*

Hányszor adódhat egymás után l hosszú növelő út?

Minden javítás után eggyel kevesebb vastag él lesz (legalább egy kritikus él törlődik).

Addig lesz l élből álló növelő út, amíg marad vastag élekből álló $s \rightsquigarrow t$ út.

Tétel. *Az Edmonds–Karp-heurisztika szerinti növelésnél a növelő utak hosszai nem csökkenő sorozatot alkotnak. Ebben a sorozatban egy adott úthosszúság legfeljebb e -szer fordulhat elő. Következésképpen legfeljebb $e(n - 1)$ növelés lehetséges. A heurisztika alkalmazásával $O(e^2 n)$ elemi lépésben kapunk maximális folyamot.*

Bonyolultabb algoritmusok

Dinic: $O(en^2)$

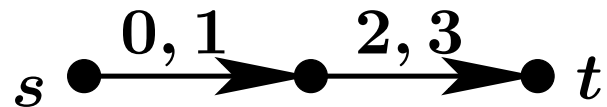
Goldberg, Tarjan: $O(en \log(n^2/e))$

Hálózatok alsó korlátokkal

Tegyük fel, hogy a $c(u, v)$ kapacitások mellett (felső korlát) *alsó korlátok* is vannak az $f(u, v)$ mennyiségekre. $\implies k(u, v) \leq f(u, v)$ is teljesüljön a G minden $u \rightarrow v$ élére.

$\implies (G, s, t, c, k)$

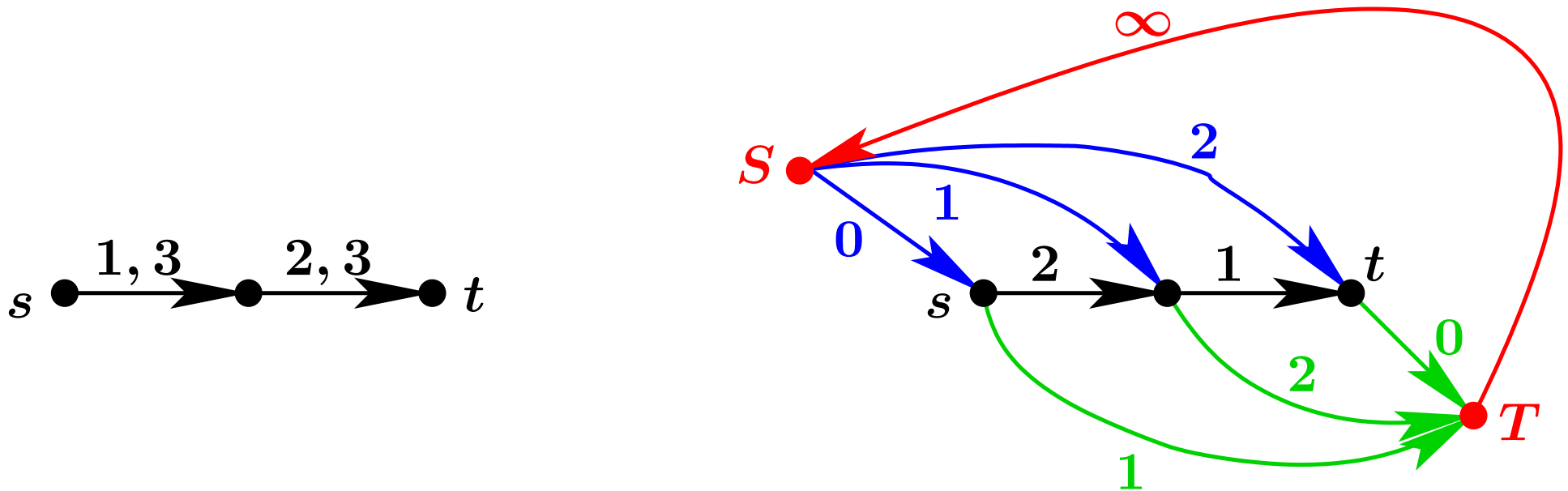
Van-e egyáltalán ilyen folyam?



Belátjuk, hogy ez a hagyományos folyamproblémára visszavezethető.

$$\mathcal{H} = (G, s, t, c, k) \rightarrow \mathcal{H}'$$

- Új forrás: S , új nyelő: T
- Régi éleken új kapacitás: $c'(u, v) := c(u, v) - k(u, v)$
- 2 új él minden pontra: $S \rightarrow v$ és $v \rightarrow T$
 $c'(S, v) := \sum_{(u,v) \in E} k(u, v)$ és $c'(v, T) := \sum_{(v,w) \in E} k(v, w)$
- Új $T \rightarrow S$ él ∞ kapacitással



Tétel. A $\mathcal{H} = (G, s, t, c, k)$ hálózatban akkor és csak akkor létezik folyam, ha a \mathcal{H}' hálózat (S -ből T -be menő) maximális folyamának az értéke $\sum_{(u,v) \in E} k(u, v)$.

Ha a $T \rightarrow S$ él kapacitást d -nek választjuk \implies ugyanígy megkaphatjuk, hogy van-e legfeljebb d értékű folyam
 \implies algoritmus alsó korlátos folyamokra

Egy ütemezési feladat

Tegyük fel, hogy egy légitársaság a J_1, J_2, \dots, J_m járatokat szeretné üzemeltetni, és d darab azonos típusú repülőgépe van erre a célra.

Minden J_i, J_j járatpárra ismert, hogy van-e elég idő arra, hogy a J_i teljesítése után egy gép felkészüljön a J_j repülésére.

Ha J_i, J_j -re a válasz igenlő, akkor azt mondjuk, hogy J_j *követheti* J_i -t.

Gráf:

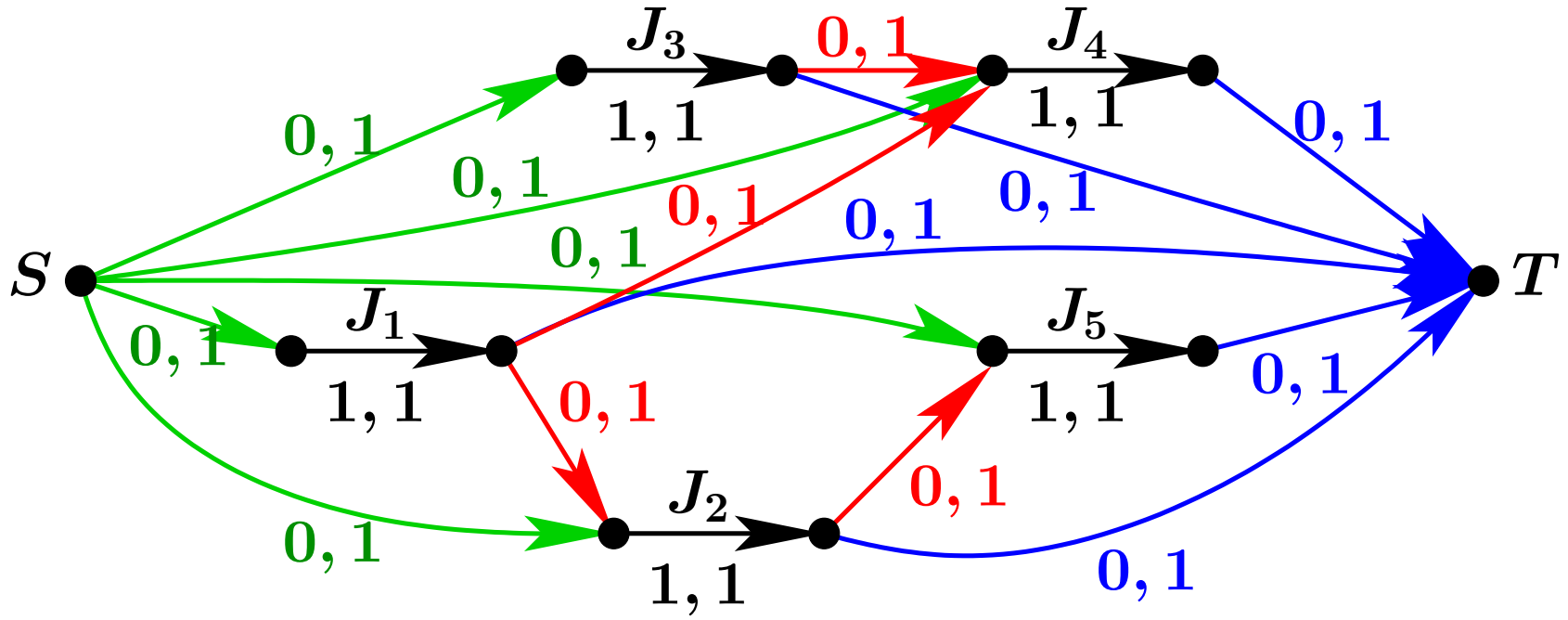
Egy J_i légijáratnak \implies két csúcs, i és i' és egy $i \rightarrow i'$ él

Ha J_j követheti J_i -t, akkor vezessünk irányított élet i' -ből j -be.

Vegyünk még fel egy s forrást és egy t nyelőt, és adjuk a hálózathoz az $s \rightarrow i$ és $i' \rightarrow t$ éleket ($1 \leq i \leq m$).

Az összes él kapacitása legyen 1. Az $i \rightarrow i'$ alakú élek alsó korlátja legyen 1, a többi élé pedig 0.

Tétel. A J_1, J_2, \dots, J_m járatok akkor és csak akkor teljesíthetők legfeljebb d géppel, ha a hálózathoz van olyan g folyam, amelyre $|g| \leq d$.



Hirdetmények

Április 1. \implies Húsvét hétfő \implies elmarad az előadás



Április 8. \implies Előadás helyett konzultáció

ZH Április 8. 16:15

A–He	CH. max.
Hi–Ka	I.B. 27
Ke–M	I.B. 28
N–Se	E.I.B.
Si–Z	St. nagy