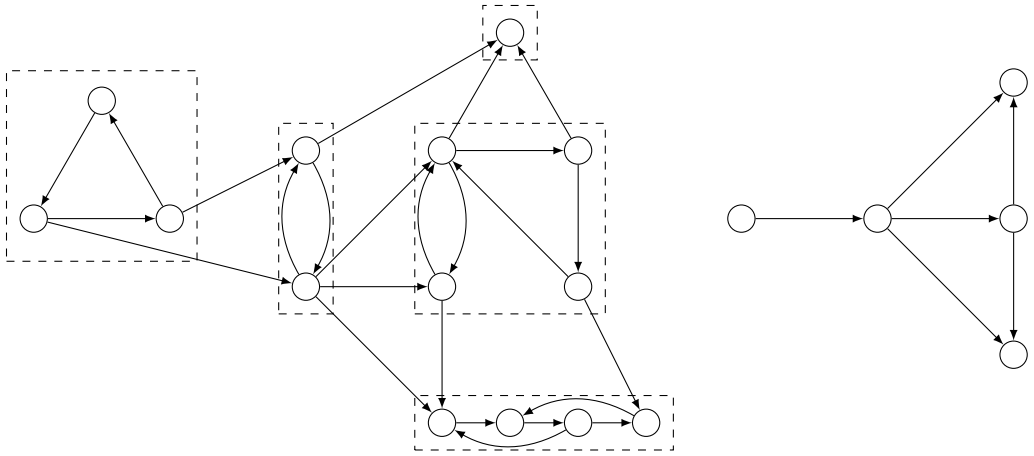
# Osme vaje APS2: sprehodi po grafih 2 (in uvod v maksimalne pretoke)

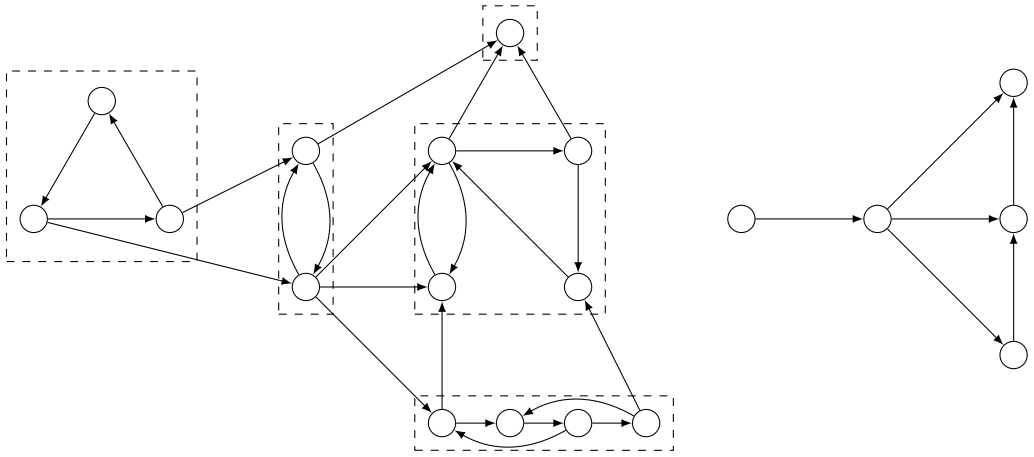
(1) Usmerjen graf je *polpovezan* natanko v primeru, če za vsak par vozlišč $u$ in $v$ obstaja vsaj ena od poti $u \rightsquigarrow v$ in $v \rightsquigarrow u$. Zasnujte algoritem za ugotavljanje polpovezanosti, ki teče v času $O(V + E)$. □

Kot vemo, lahko usmerjen graf $G = (V, E)$ razstavimo na množico krepko povezanih komponent. Krepko povezana komponenta je maksimalna (torej nerazširljiva) množica $C \subseteq V$, tako da za vsak par $u, v \in C$ obstaja tako pot $u \rightsquigarrow v$ kot pot $v \rightsquigarrow u$. Ker so posamezne krepko povezane komponente po definiciji polpovezane, nas zanimajo samo povezave *med* komponentami. To pomeni, da se lahko omejimo na *komponentni graf*, v katerem so vozlišča posamezne komponente, povezava $(C_i, C_j)$ pa obstaja natanko v primeru, če obstajata vozlišči $u \in C_i$ in $v \in C_j$, tako da je $(u, v) \in E$. Graf $G$ bo torej polpovezan natanko tedaj, ko bo tak tudi njegov komponentni graf.

Za lažjo predstavo sta na slikah 1 in 2 prikazana dva grafa in pripadajoča komponentna grafa. Prvi graf ni polpovezan, drugi pa je.



Slika 1: Graf, ki ni polpovezan, in njegov komponentni graf.



Slika 2: Polpovezan graf in njegov komponentni graf.

Komponentni graf je seveda acikličen; če bi obstajal cikel $C_{i_1} \to C_{i_2} \to \ldots \to C_{i_k} \to C_{i_1}$, potem bi vse te »komponente« v resnici bile ena sama komponenta. No, če imamo

usmerjen acikličen graf, potem lahko zanj zgradimo ... topološki vrstni red! Pa nam to res kakorkoli pomaga pri ugotavljanju polpovezanosti?

Izkaže se, da je odgovor DA, vendar pa je bolje razmišljati o klasičnem algoritmu za topološko urejanje, ki ste ga spoznali pri APS 1, kot o algoritmu, ki temelji na DFS. Klasični algoritem preprosto odstranjuje vozlišča brez vstopnih povezav in tako neposredno zgradi topološki vrstni red. No, če vzamemo kos papirja in naredimo nekaj primerov, bomo prej ali slej prišli do sledeče ugotovitve:

**Trditev 1.** *Usmerjen acikličen graf je polpovezan natanko tedaj, ko zanj obstaja natanko en topološki vrstni red.*

*Dokaz.* ($\Rightarrow$) Dokazali bomo kontrapozicijo: če imamo več topoloških vrstnih redov, potem graf ni polpovezan.

Kdaj obstaja več topoloških vrstnih redov? Takrat, ko imamo v nekem koraku izvajanja klasičnega algoritma za topološko urejanje na voljo več možnih vozlišč. Recimo, da se nam je to pravkar zgodilo in da sta kandidata za naslednje mesto v topološkem vrstnem redu vozlišči $u$ in $v$. Ali obstaja pot od $u$ do $v$? Ne, ker če bi ta pot obstajala, bi $v$ v topološkem vrstnem redu obvezno nastopal kasneje kot $u$. Ali obstaja pot od $v$ do $u$? Takisto ne — iz simetričnega razloga! Ergo: graf ni polpovezan.

($\Leftarrow$) Naj bo $v_1 \prec v_2 \prec \ldots \prec v_n$ enolično določen topološki vrstni red. Zadošča, da pokažemo, da v grafu obstaja pot $v_1 \to v_2 \to \ldots \to v_n$; če je to res, bo za poljuben par vozlišč $v_i$ in $v_j$ $(i < j)$ obstajala pot $v_i \to v_{i+1} \to \ldots \to v_j$.

Trditev dokažimo z indukcijo po dolžini poti ($k$). Za $k = 1$ pot seveda vedno obstaja, saj je sestavljena iz enega samega vozlišča ($v_1$). Sedaj pa predpostavimo, da obstaja pot $v_1 \to v_2 \to \ldots \to v_{k-1}$, in razmislimo, ali obstaja tudi povezava $v_{k-1} \to v_k$. No, če ne bi obstajala, potem bi v trenutku, ko bi klasični algoritem »ogolil« (torej odstranil vse vstopne povezave) vozlišče $v_{k-1}$, ogolil tudi vozlišče $v_k$. Zakaj? Hja, povezave $v_\ell \to v_k$ za $\ell > k$ ne morejo obstajati, ker $v_\ell$ po predpostavki v topološkem vrstnem redu nastopa kasneje kot $v_k$, povezave $v_\ell \to v_k$ za $\ell < k - 1$ pa so nekoč seveda lahko obstajale, a jih je algoritem do tega trenutka že odstranil. Če torej povezava $v_{k-1} \to v_k$ prav tako ne obstaja, potem bo v trenutku, ko algoritem ogoli vozlišče $v_{k-1}$, golo tudi vozlišče $v_k$, to pa pomeni, da imamo v tistem hipu dva enakovredna topološka vrstna reda.

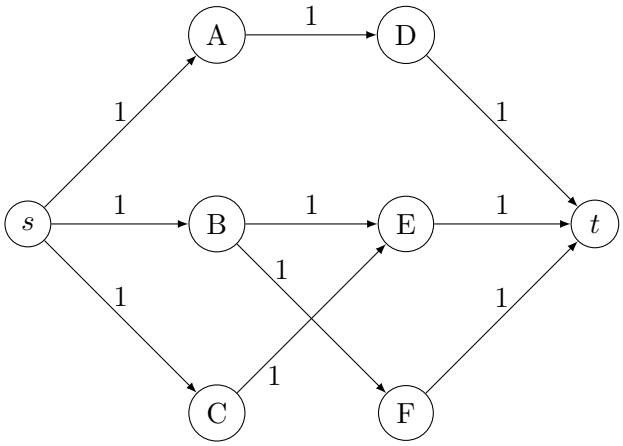
□

To je to! V času $O(V + E)$ izvršimo algoritem za odkrivanje krepko povezanih komponent, da nam zgradi komponentni graf, in v času $O(V + E)$ na komponentnem grafu izvršimo klasični algoritem za topološko urejanje. Če kadarkoli ogolimo več kot eno vozlišče, vemo, da graf ni polpovezan, če se nam to nikoli ne zgodi, pa zaključimo, da je.

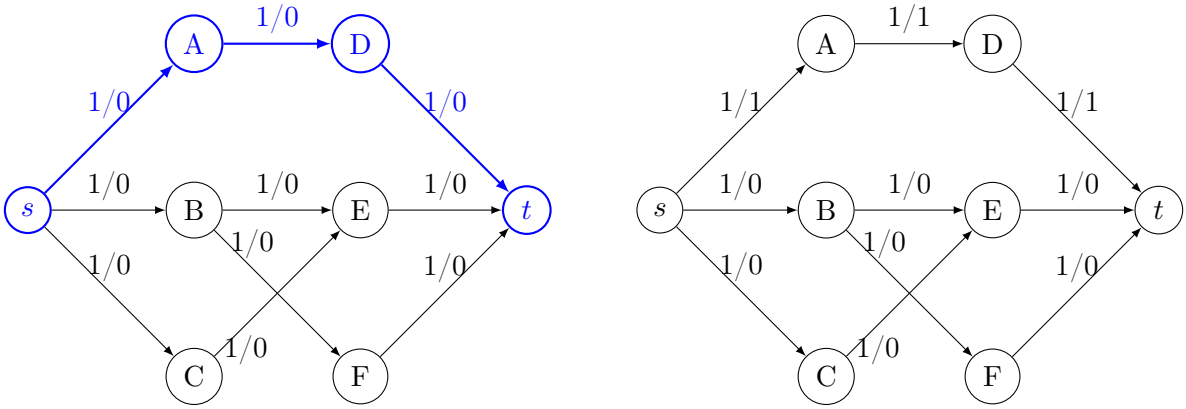
(2) Določite maksimalni pretok za omrežje na sliki 3. □

Nič lažjega! V prvi iteraciji (slika 4) najdemo, recimo, nezasičeno pot $s \to \mathrm{A} \to \mathrm{D} \to t$ in jo zasitimo. V drugi iteraciji (slika 5) najdemo nezasičeno pot $s \to \mathrm{B} \to \mathrm{E} \to t$ in jo zasitimo. Na prvi pogled je zgodba končana, vendar pa lahko najdemo še eno nezasičeno pot (slika 6): $s \to \mathrm{C} \to \mathrm{E} \leftarrow \mathrm{B} \to \mathrm{F} \to t$. Da, pri pretokih lahko povezave obravnavamo v obeh smereh; povezava $(\mathrm{B}, \mathrm{E})$ v smeri $\mathrm{B} \to \mathrm{E}$ nima nobene rezerve, v smeri $\mathrm{E} \to \mathrm{B}$ pa ima eno enoto rezerve. Pretok po njej lahko zato *zmanjšamo* za eno

enoto in ga na ta račun drugod povečamo za eno enoto. Končna vrednost pretoka je torej 3.




Slika 3: Primer omrežja.

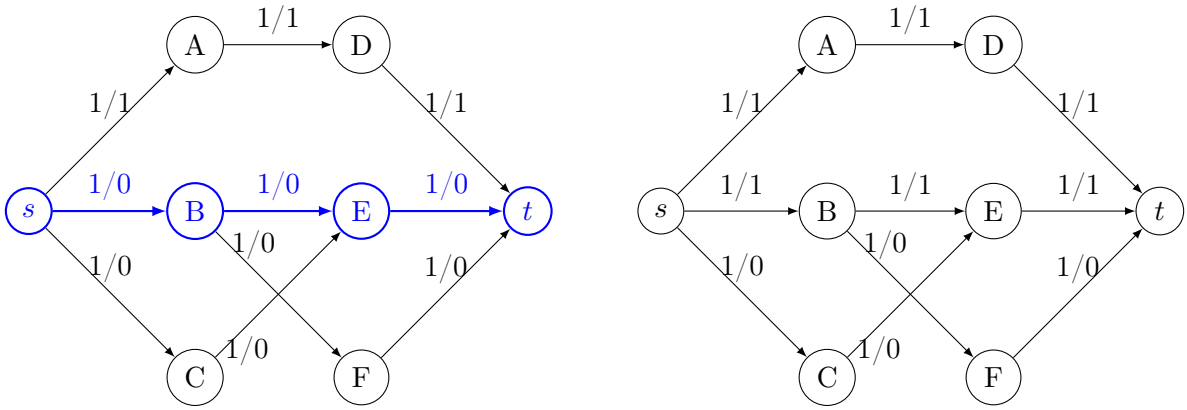



Slika 4: Prva iteracija.

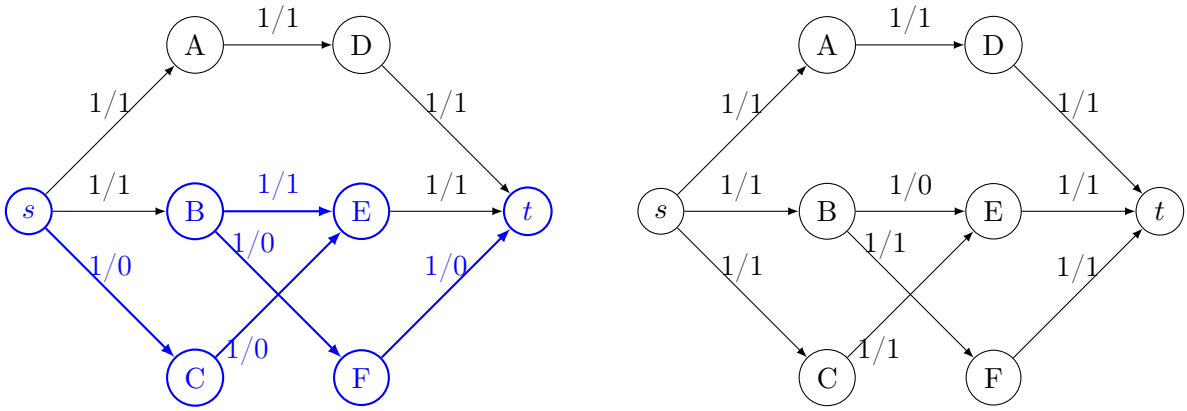
(3) Na predavanjih smo se pri iskanju maksimalnega pretoka po omrežju omejili na primere, ko med vsakim parom vozlišč $u$ in $v$ obstaja kvečjemu ena od povezav $(u, v)$ in $(v, u)$. Ta omejitev je sicer priročna, ker nam poenostavi kak dokaz ali dva, vendar pa ni potrebna. Cormen *et al.* ponujajo preprost recept za obravnavo takih primerov: če imamo v grafu poleg povezave $(u, v)$ še povezavo $(v, u)$ (recimo, da ima kapaciteto $c$), jo nadomestimo z novim vozliščem $w$ ter povezavama $(v, w)$ in $(w, u)$ s kapaciteto $c$.

Izkaže pa se, da niti to ni potrebno: dvosmerne povezave lahko obravnavamo povsem enako kot enosmerne, upoštevamo le, da pretok vedno teče zgolj v eni smeri. Recimo, da je kapaciteta povezave $(u, v)$ enaka $c$, kapaciteta povezave $(v, u)$ pa $d$ in da trenutno teče $x \leqslant c$ enot v smeri $u \to v$. Pretok v smeri $v \to u$ lahko sedaj povečamo za $x + d$ enot ($x$ enot za nevtralizacijo pretoka v smeri $i \to j$ in še $d$ enot, ki jih dopušča povezava $v \to u$).

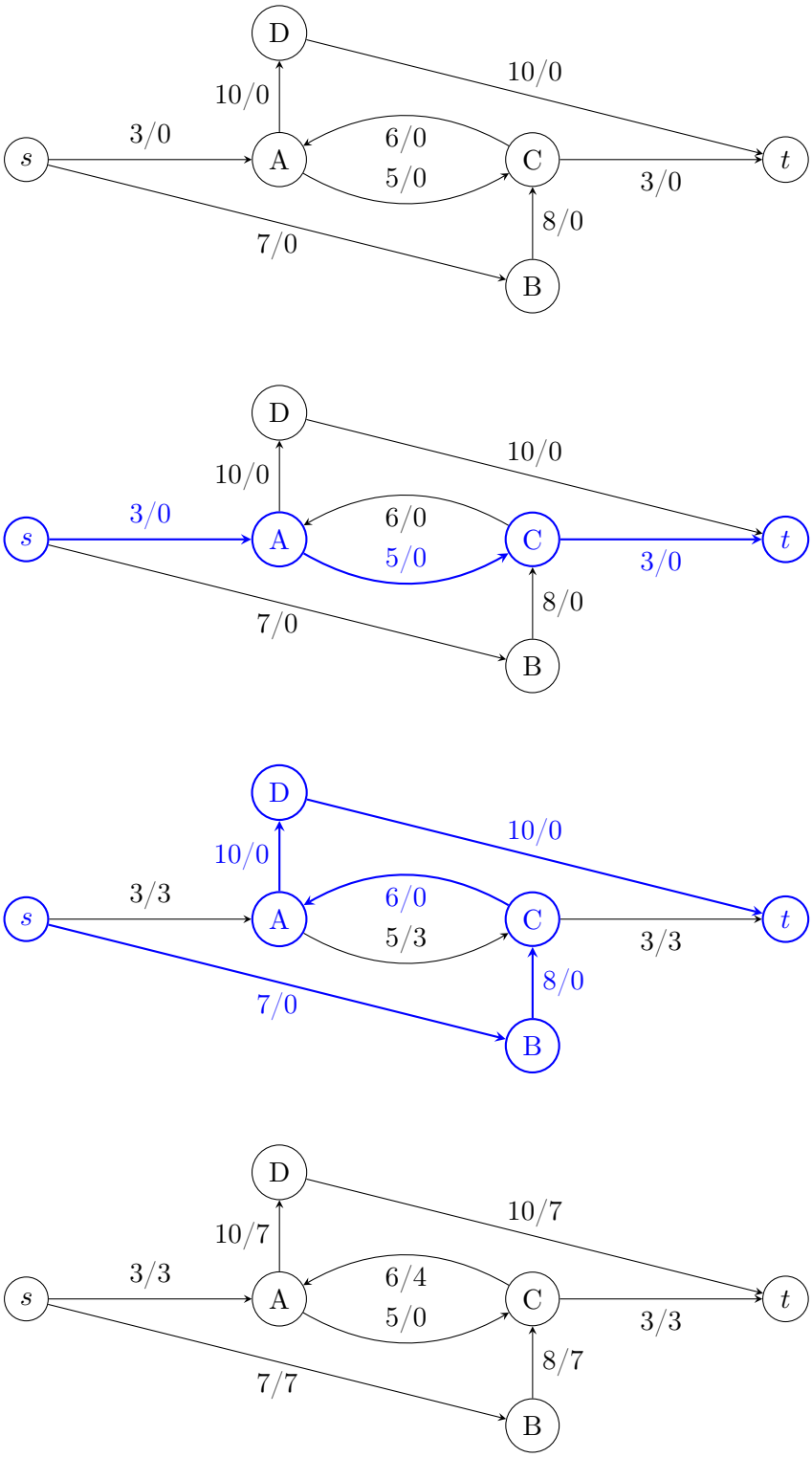
Oglejmo si primer na sliki 7. V prvi iteraciji zasitimo pot $s \to \mathrm{A} \to \mathrm{C} \to t$, v drugi pa odkrijemo nezasičeno pot $s \to \mathrm{B} \to \mathrm{C} \to \mathrm{A} \to \mathrm{D} \to t$. Povezavi $\mathrm{C} \to \mathrm{A}$, po kateri trenutno tečejo 3 enote v nasprotni smeri, bi lahko pretok povečali celo za 9 enot (3 enote za nevtralizacijo pretoka v nasprotni smeri in še 6 enot za zapolnitev kapacitete), vendar pa smo omejeni s 7 enotami, ki jih s seboj prinesemo iz vozlišča $s$.




Slika 5: Druga iteracija.




Slika 6: Tretja iteracija.




Slika 7: Omrežje z dvosmerno povezavo.