Stabla

Diskretne strukture s teorijom grafova

Damir Horvat

FOI, Varaždin

Zadatak 1

U stablu T je $\Delta(T) = 5$. Pritom, stablo T ima četiri vrha stupnja 2, jedan vrh stupnja 3, dva vrha stupnja 4 i jedan vrh stupnja 5. Koliko listova ima stablo T i koliki je ukupni broj vrhova i bridova u promatranom stablu?

Rješenje

$\ell \longleftarrow$ broj listova u stablu T

$\Delta(T) = 5, \quad \nu = \ell + 8, \quad \varepsilon = \nu - 1$

$$\sum_{v \in V(T)} d(v) = 2\varepsilon$$

$$\ell \cdot 1 + 4 \cdot 2 + 1 \cdot 3 + 2 \cdot 4 + 1 \cdot 5 = 2(\nu - 1)$$

$$\ell + 24 = 2(\ell + 8 - 1)$$

$$\ell + 24 = 2(\ell + 7)$$

$$\ell + 24 = 2\ell + 14$$

$$\boxed{\ell = 10}$$

$$\nu = \ell + 8$$

$$\nu = 10 + 8$$

$$\boxed{\nu = 18}$$

$$\varepsilon = \nu - 1$$

$$\varepsilon = 18 - 1$$

$$\boxed{\varepsilon = 17}$$

Teorem (rekurzija za broj razapinjućih stabala)

Neka je $\tau(G)$ broj razapinjućih stabala grafa G. Ako je $e \in E(G)$ karika, tada vrijedi

$$\tau(G) = \tau(G - e) + \tau(G \cdot e).$$

Matrični teorem o stablima

Neka je G povezani graf bez petlji sa skupom vrhova $V = \{v_1, v_2, \dots, v_n\}$. Neka je $A = [a_{ij}]$ matrica susjedstva grafa G. Neka je $Q = [q_{ij}]$ $n \times n$ matrica za koju vrijedi

$$q_{ij} = \begin{cases} -a_{ij}, & i \neq j \\ d_G(v_i), & i = j \end{cases}.$$

Tada je $\tau(G)$ jednak bilo kojem kofaktoru matrice Q.

Zadatak 2

Graf G zadan je matricom susjedstva

$$A = \begin{bmatrix} 1 & 3 & 0 & 0 \\ 3 & 0 & 1 & 1 \\ 0 & 1 & 2 & 2 \\ 0 & 1 & 2 & 0 \end{bmatrix}.$$

a) *Bez crtanja grafa G odredite ukupni broj petlji u grafu i stupnjeve njegovih vrhova.*

b) *Nacrtajte graf G i pomoću rekurzije odredite ukupni broj razapinjućih stabala grafa G.*

c) *Pomoću matričnog teorema o stablima odredite ukupni broj razapinjućih stabala grafa G.*

d) *Je li G Hamiltonov graf? Postoji li Hamiltonov put u grafu G?*

Rješenje

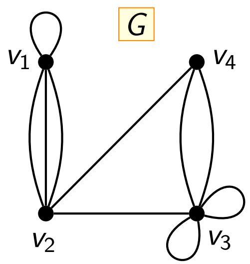
a)

$$A = \begin{array}{c|cccc} & v_1 & v_2 & v_3 & v_4 \\ \hline v_1 & 1 & 3 & 0 & 0 \\ v_2 & 3 & 0 & 1 & 1 \\ v_3 & 0 & 1 & 2 & 2 \\ v_4 & 0 & 1 & 2 & 0 \end{array}$$

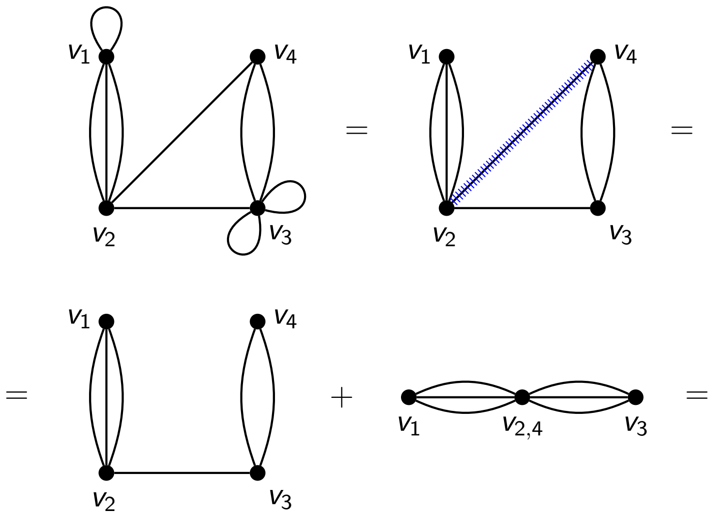
$$\text{broj petlji} = 1 + 0 + 2 + 0 = 3$$

$$d(v_1) = 2 \cdot 1 + 3 + 0 + 0 = 5$$
$$d(v_2) = 3 + 2 \cdot 0 + 1 + 1 = 5$$
$$d(v_3) = 0 + 1 + 2 \cdot 2 + 2 = 7$$
$$d(v_4) = 0 + 1 + 2 + 2 \cdot 0 = 3$$

b)



$$A = \begin{array}{c|cccc} & v_1 & v_2 & v_3 & v_4 \\ \hline v_1 & 1 & 3 & 0 & 0 \\ v_2 & 3 & 0 & 1 & 1 \\ v_3 & 0 & 1 & 2 & 2 \\ v_4 & 0 & 1 & 2 & 0 \end{array}$$



$= 3 \cdot 1 \cdot 2 + 3 \cdot 3 = 15$ $\boxed{\tau(G) = 15}$

$\boxed{\tau(G) = \tau(G - e) + \tau(G \cdot e)}$

$\boxed{A_{ij} = (-1)^{i+j} M_{ij}}$

c) Najprije treba ukloniti sve petlje iz grafa G.

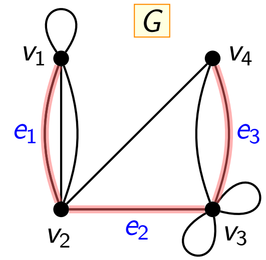
$$A = \begin{bmatrix} 1 & 3 & 0 & 0 \\ 3 & 0 & 1 & 1 \\ 0 & 1 & 2 & 2 \\ 0 & 1 & 2 & 0 \end{bmatrix} \quad A_1 = \begin{bmatrix} 0 & 3 & 0 & 0 \\ 3 & 0 & 1 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 2 & 0 \end{bmatrix} \quad Q = \begin{bmatrix} 3 & -3 & 0 & 0 \\ -3 & 5 & -1 & -1 \\ 0 & -1 & 3 & -2 \\ 0 & -1 & -2 & 3 \end{bmatrix}$$

$$Q_{21} = (-1)^{2+1} \cdot \begin{vmatrix} -3 & 0 & 0 \\ -1 & 3 & -2 \\ -1 & -2 & 3 \end{vmatrix} = -1 \cdot (-15) = 15$$

$\boxed{\tau(G) = 15}$

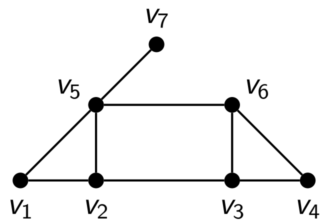
d) G ima Hamiltonov put, npr. $v_1 e_1 v_2 e_2 v_3 e_3 v_4$.

G nije Hamiltonov graf jer je v_1 problematični vrh.



Zadatak 3

Zadan je graf G.



a) *Pomoću rekurzije odredite ukupni broj razapinjućih stabala grafa G.*

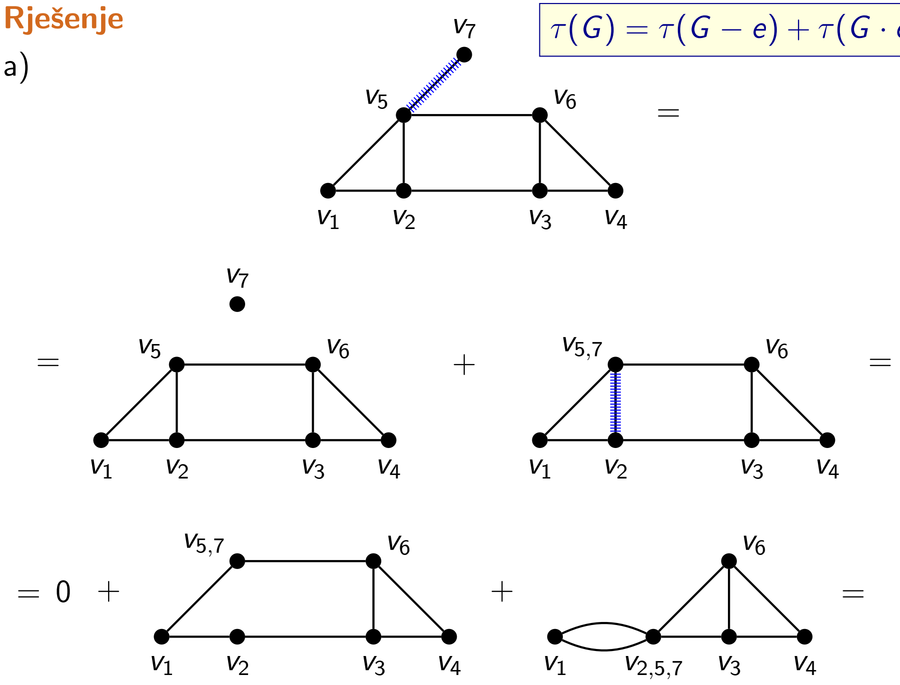
b) *Pomoću BFS algoritma pronađite jedno razapinjuće stablo grafa G.*

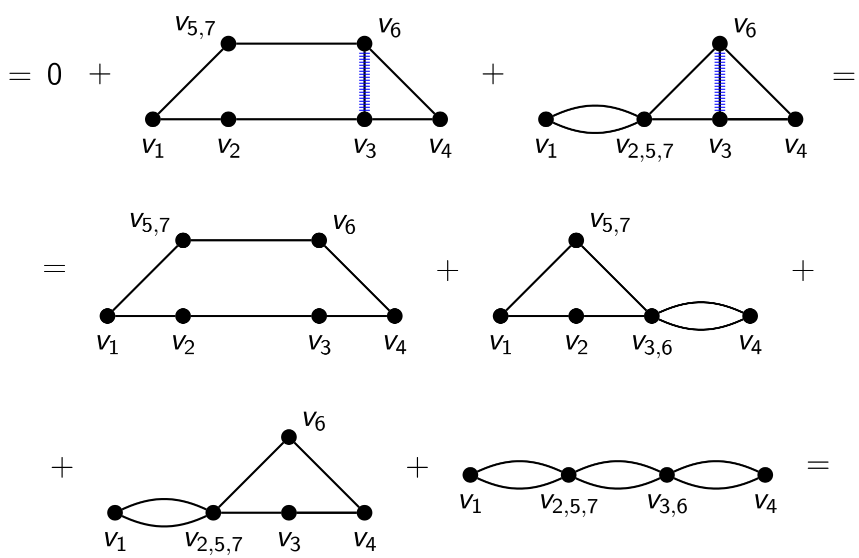
c) *Pomoću DFS algoritma pronađite jedno razapinjuće stablo grafa G.*

Rješenje

a)

$$\tau(G) = \tau(G-e) + \tau(G \cdot e)$$





$$= 6 + 4 \cdot 2 + 2 \cdot 4 + 2 \cdot 2 \cdot 2 = 30$$

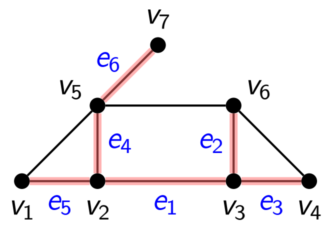
$$\boxed{\tau(G) = 30}$$

$$\boxed{\tau(G) = \tau(G-e) + \tau(G \cdot e)}$$

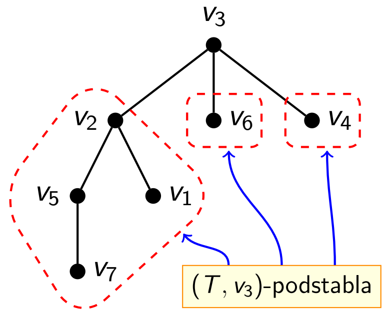
Implementacija BFS algoritma

```
procedure BFS(G, s)
    for u ∈ V(G) \ {s} do                ▷ inicijalizacija za sve vrhove osim vrha s
        color[u] ← WHITE                 ▷ vrhovi su bijele boje
        d[u] ← ∞                         ▷ vrhovi su na udaljenosti ∞ od vrha s
        π[u] ← NIL                       ▷ vrhovi nemaju roditelje
    color[s] ← GRAY                      ▷ vrh s je sive boje (vrh je posjećen, ali nije istražen)
    d[s] ← 0                             ▷ vrh s je na udaljenosti 0 od samoga sebe
    π[s] ← NIL                           ▷ vrh s nema roditelja
    Q ← ∅                                ▷ inicijalizacija reda Q koji će se puniti sa sivim vrhovima
    ENQUEUE(Q, s)                        ▷ sivi vrh s stavi na kraj reda Q
    while Q ≠ ∅ do                       ▷ sve dok ima sivih vrhova
        u ← DEQUEUE(Q)                   ▷ uzmi sivi vrh u koji je prvi u redu Q
        for v ∈ Adj[u] do                ▷ za svaki susjedni vrh v od vrha u
            if color[v] = WHITE then     ▷ ako je vrh v bijele boje
                color[v] ← GRAY          ▷ pridruži vrhu v sivu boju
                d[v] ← d[u] + 1          ▷ udaljenost v od s je za 1 veća od udaljenosti u od s
                π[v] ← u                 ▷ vrh u je roditelj vrha v
                ENQUEUE(Q, v)            ▷ stavi sivi vrh v na kraj reda Q
        color[u] ← BLACK                 ▷ pridruži vrhu u crnu boju (vrh u je istražen)
```


b)

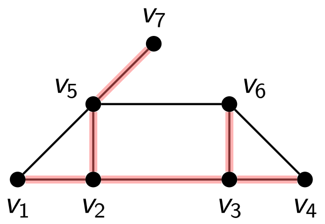


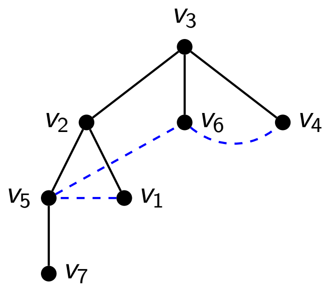
korak	BFS
1	$e_1, \ u = v_3$
2	$e_1 e_2, \ u = v_3$
3	$e_1 e_2 e_3, \ u = v_3$
4	$e_1 e_2 e_3, \ u = v_2$
5	$e_1 e_2 e_3 e_4, \ u = v_2$
6	$e_1 e_2 e_3 e_4 e_5, \ u = v_2$
7	$e_1 e_2 e_3 e_4 e_5, \ u = v_6$
8	$e_1 e_2 e_3 e_4 e_5, \ u = v_4$
9	$e_1 e_2 e_3 e_4 e_5, \ u = v_5$
10	$e_1 e_2 e_3 e_4 e_5 e_6, \ u = v_5$



	$\pi(v)$	$d(v)$
v_1	v_2	2
v_2	v_3	1
v_3	$-$	0
v_4	v_3	1
v_5	v_2	2
v_6	v_3	1
v_7	v_5	3

Neka svojstva BFS algoritma





Neka je G povezani graf i (T, v) korijensko stablo dobiveno BFS algoritmom započetom u vrhu $v \in V(G)$.

- Krajevi svakog brida povezanog grafa G koji ne pripada BFS stablu (T, v) nalaze se na istom ili susjednim nivoima stabla (T, v).

Neka svojstva BFS algoritma

- U povezanom grafu G postoji ciklus neparne duljine akko postoji brid iz $E(G) \setminus E(T)$ koji ima oba kraja na istom nivou BFS stabla (T, v).
- U povezanom grafu G postoji ciklus koji sadrži vrh $v \in V(G)$ akko postoji brid iz $E(G) \setminus E(T)$ čiji krajevi pripadaju različitim (T, v)-podstablima BFS stabla (T, v).
- U povezanom grafu G postoji ciklus koji sadrži brid $\{v, w\}$ akko postoji brid iz $E(G) \setminus E(T)$ kojemu jedan kraj pripada (T, v)-podstablu s korijenom w, a drugi kraj pripada nekom drugom (T, v)-podstablu.

Modifikacije BFS algoritma

Nivoi u BFS stablu

Standardni BFS algoritam pokrenut u vrhu $v \in V(G)$ daje udaljenosti $d(y)$ vrha v do svakog vrha $y \in V(G)$.

Pomoću vrijednosti $\pi(y)$ rekonstruiraju se najkraći putovi od vrha v do svih preostalih vrhova u grafu G.

Za svaki brid $\{x, y\}$ koji ne pripada BFS stablu (T, v), ispitivanjem uvjeta $d(x) = d(y)$ dobivamo algoritam koji daje odgovor na pitanje *"Postoji li u grafu G ciklus neparne duljine?"*.

Modifikacije BFS algoritma

BFS podstabla

Pretpostavimo da unutar BFS algoritma pokrenutog iz vrha $v \in V(G)$ za svaki vrh $y \in V(G)$ spremamo informaciju $\alpha(y)$ kojem (T, v)-podstablu vrh y pripada.

Za svaki brid $\{x, y\}$ koji ne pripada BFS stablu (T, v), istraživanjem vrijednosti $\alpha(x)$ i $\alpha(y)$ dobivamo

- algoritam koji određuje postoji li u grafu G ciklus koji sadrži vrh v,
- algoritam koji određuje postoji li u grafu G ciklus koji sadrži brid $\{v, w\}$.

Modifikacije BFS algoritma

BFS nivoi i BFS podstabla

Pretpostavimo da unutar BFS algoritma pokrenutog iz vrha $v \in V(G)$ za svaki vrh $y \in V(G)$ spremamo informacije $d(y)$ i $\alpha(y)$.

Za svaki brid $\{x, y\}$ koji ne pripada BFS stablu (T, v), istraživanjem vrijednosti $\alpha(x)$ i $\alpha(y)$ dobivamo

- algoritam koji određuje duljinu najkraćeg ciklusa u grafu G koji sadrži vrh v,
- algoritam koji određuje duljinu najkraćeg ciklusa u grafu G koji sadrži brid $\{v, w\}$.

Modifikacije BFS algoritma

Nakon što prvi put pronađemo brid $\{x, y\} \in E(G) \setminus E(T)$ za koji je $\alpha(x) \neq \alpha(y)$, tada uspoređujemo vrijednosti $d(x)$ i $d(y)$.

Ako je $d(x) = d(y)$, tada algoritam završava.

Ako je $d(x) \neq d(y)$, npr. $d(x) < d(y)$, tada moramo nastaviti s istraživanjem vrhova koji se nalaze na nivou $d(x)$ kako bismo saznali postoji li brid koji ima oba kraja na nivou $d(x)$ koji pripadaju različitim (T, v)-podstablima.

Dakle, ako postoji ciklus u grafu G koji sadrži vrh v, duljina najkraćeg takvog ciklusa jednaka je $2d(x) + 1$ ili $2d(x) + 2$.

Modifikacije BFS algoritma

Određivanje struka grafa

Ako primijenimo algoritam za određivanje najkraćeg ciklusa u grafu G koji sadrži vrh v na svaki vrh $v \in V(G)$, dobivamo algoritam za određivanje struka grafa koji ima složenost $O(\nu\varepsilon)$.

Ako pritom još za svaki vrh $v \in V(G)$ spremamo informaciju $\pi(v)$, možemo rekonstruirati jedan takav najkraći ciklus.

Neki problemi su ipak teški

Odrediti ciklus neparne duljine u grafu G koji sadrži vrh v.

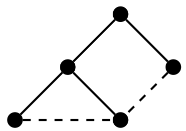
- Ovaj problem je u općenitom slučaju jednako težak kao i problem ispitivanja je li graf Hamiltonov.
- Problem je težak jer nemamo općenitu tvrdnju koja bi davala nužne i dovoljne uvjete za postojanje neparnog ciklusa u grafu G koji sadrži zadani vrh v.

Neki problemi su ipak teški

Tvrdnja 1.

Ako postoji brid $e \in E(G) \setminus E(T)$ koji ima oba kraja na istom nivou BFS stabla (T, v) i u različitim (T, v)-podstablima, tada u grafu G postoji neparni ciklus koji sadrži vrh v.

Gornja tvrdnja daje samo dovoljan, ali ne i nužan uvjet za postojanje neparnog ciklusa koji sadrži zadani vrh u grafu.

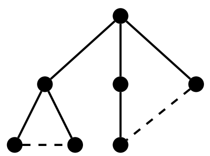


Neki problemi su ipak teški

Tvrdnja 2.

Ako u grafu G postoji neparni ciklus koji sadrži vrh v, tada postoji brid $e_1 \in E(G) \setminus E(T)$ koji ima oba kraja na istom nivou u BFS stablu (T, v) i postoji brid $e_2 \in E(G) \setminus E(T)$ čiji krajevi pripadaju različitim (T, v)-podstablima.

Gornja tvrdnja daje samo nužan, ali ne i dovoljan uvjet za postojanje neparnog ciklusa koji sadrži zadani vrh u grafu.



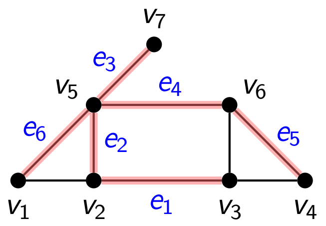
Rekurzivna implementacija DFS algoritma

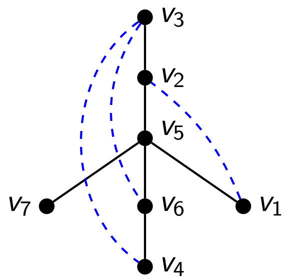
```
procedure DFS(G)
  for u ∈ V(G) do                     ▷ inicijalizacija za sve vrhove grafa G
    color[u] ← WHITE                  ▷ vrhovi su bijele boje
    π[u] ← NIL                        ▷ vrhovi nemaju roditelje
  time ← 0                            ▷ globalna varijabla koja mjeri vrijeme
  for u ∈ V(G) do                     ▷ za svaki vrh u grafa G
    if color[u] = WHITE then          ▷ ako je vrh u bijele boje
      DFS_VISIT(u)                    ▷ posjeti vrh u s DFS_VISIT procedurom
procedure DFS_VISIT(u)
  color[u] ← GRAY                     ▷ pridruži vrhu u sivu boju
  time ← time + 1                     ▷ povećaj vrijeme za 1
  d[u] ← time                         ▷ spremanje trenutka kada je vrh u otkriven
  for v ∈ Adj[u] do                   ▷ za svaki susjedni vrh v od vrha u
    if color[v] = WHITE then          ▷ ako je vrh v bijele boje
      π[v] ← u                        ▷ vrh u je roditelj vrha v
      DFS_VISIT(v)                    ▷ posjeti vrh v s DFS_VISIT procedurom
  color[u] ← BLACK                    ▷ pridruži vrhu u crnu boju (vrh u je istražen)
  time ← time + 1                     ▷ povećaj vrijeme za 1
  f[u] ← time                         ▷ spremanje trenutka kada je vrh u istražen
```


Implementacija DFS algoritma pomoću stoga

```
procedure DFS_VISIT(u)
  S ← ∅                               ▷ inicijalizacija stoga koji će se puniti s vrhovima grafa G
  PUSH(S, u)                          ▷ stavi vrh u na stog S
  while S ≠ ∅ do                      ▷ sve dok na stogu S ima vrhova
    x ← POP(S)                        ▷ uzmi vrh sa stoga S i spremi ga u varijablu x
    if color[x] = WHITE then          ▷ ako je vrh x bijele boje
      time ← time + 1                 ▷ povećaj vrijeme za 1
      d[x] ← time                     ▷ spremanje trenutka kada je vrh x otkriven
      color[x] ← GRAY                 ▷ pridruži vrhu x sivu boju
      PUSH(S, x)                      ▷ stavi ponovo vrh x na stog S
      for v ∈ Adj[x] do               ▷ za svaki susjedni vrh v od vrha x
        if color[v] = WHITE then      ▷ ako je vrh v bijele boje
          π[v] ← x                    ▷ stavi da je vrh x roditelj vrha v
          PUSH(S, v)                  ▷ stavi vrh v na stog S
    else if color[x] = GRAY then      ▷ ako vrh x nije bijele boje, ali je sive boje
      time ← time + 1                 ▷ povećaj vrijeme za 1
      f[x] ← time                     ▷ spremanje trenutka kada je vrh x istražen
      color[x] ← BLACK                ▷ pridruži vrhu x crnu boju (vrh x je istražen)
```

c)





korak	DFS
1	$e_1, \ u = v_2$
2	$e_1 e_2, \ u = v_5$
3	$e_1 e_2 e_3, \ u = v_7$
4	$e_1 e_2 e_3, \ u = v_5$
5	$e_1 e_2 e_3 e_4, \ u = v_6$
6	$e_1 e_2 e_3 e_4 e_5, \ u = v_4$
7	$e_1 e_2 e_3 e_4 e_5, \ u = v_6$
8	$e_1 e_2 e_3 e_4 e_5, \ u = v_5$
9	$e_1 e_2 e_3 e_4 e_5 e_6, \ u = v_1$
	$u = v_5, \ u = v_2, \ u = v_3$

	$\pi(v)$	$d(v)$	$f(v)$
v_1	v_5	10	11
v_2	v_3	2	13
v_3	—	1	14
v_4	v_6	7	8
v_5	v_2	3	12
v_6	v_5	6	9
v_7	v_5	4	5

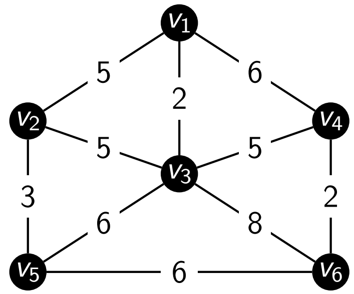
Teorem

Ako primijenimo DFS algoritam na neusmjereni ili usmjereni graf G, tada za svaka dva vrha $v_1, v_2 \in V(G)$ vrijedi točno jedan od sljedeća tri uvjeta:

- *Intervali $\big[d(v_1), f(v_1)\big]$ i $\big[d(v_2), f(v_2)\big]$ su disjunktni. Vrh v_1 nije potomak vrha v_2 i vrh v_2 nije potomak vrha v_1.*
- *Interval $\big[d(v_1), f(v_1)\big]$ je sadržan unutar intervala $\big[d(v_2), f(v_2)\big]$. Vrh v_1 je potomak vrha v_2 u pripadnoj DFS šumi.*
- *Interval $\big[d(v_2), f(v_2)\big]$ je sadržan unutar intervala $\big[d(v_1), f(v_1)\big]$. Vrh v_2 je potomak vrha v_1 u pripadnoj DFS šumi.*

Zadatak 4

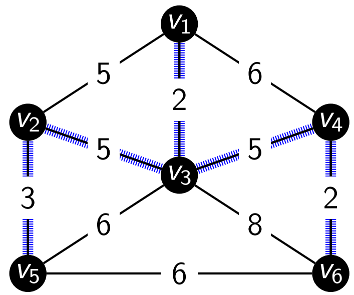
Pomoću Kruskalovog i Primovog algoritma pronađite minimalno razapinjuće stablo u težinskom grafu G.



Rješenje

Kruskal

težina stabla: $2 + 2 + 3 + 5 + 5 = 17$



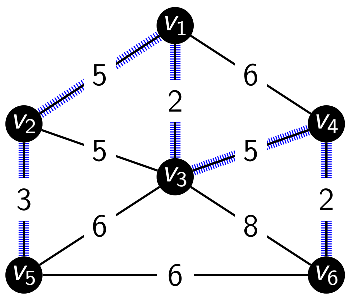
korak	1	2	3	4	5
brid	$v_1 v_3$	$v_4 v_6$	$v_2 v_5$	$v_2 v_3$	$v_3 v_4$
težina	2	2	3	5	5

Rješenje

Prim

težina stabla: $3+5+2+5+2=17$

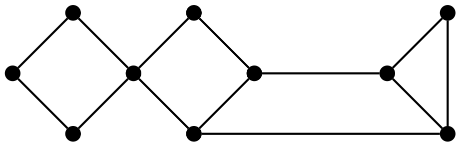
početni vrh: v_5



korak	1	2	3	4	5
brid	v_2v_5	v_1v_2	v_1v_3	v_3v_4	v_4v_6
težina	3	5	2	5	2

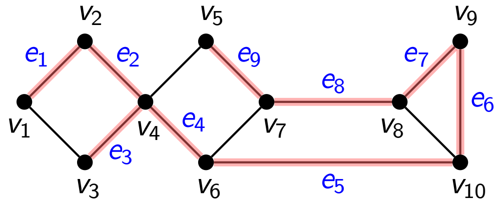
Zadatak 5

Odredite jednu jaku orijentaciju na grafu G ukoliko ona postoji.



Rješenje

Jaka orijentacija na grafu G postoji jer graf G nema reznih bridova.



korak	DFS
1	$e_1, \ u=v_2$
2	$e_1e_2, \ u=v_4$
3	$e_1e_2e_3, \ u=v_3$
4	$e_1e_2e_3, \ u=v_4$
5	$e_1e_2e_3e_4, \ u=v_6$
6	$e_1e_2e_3e_4e_5, \ u=v_{10}$
7	$e_1e_2e_3e_4e_5e_6, \ u=v_9$
8	$e_1e_2e_3e_4e_5e_6e_7, \ u=v_8$
9	$e_1e_2e_3e_4e_5e_6e_7e_8, \ u=v_7$
10	$e_1e_2e_3e_4e_5e_6e_7e_8e_9, \ u=v_5$