

Univerza v Ljubljani  
Fakulteta za računalništvo in informatiko  
Tržaška cesta 25, Ljubljana

Zoran Bosnić

# **PROGRAMIRANJE IN ALGORITMI**

algoritmi za analizo grafov

Ljubljana, september 2010

---

BOSNIĆ, Zoran

Programiranje in algoritmi: algoritmi za analizo grafov  
Ljubljana : Fakulteta za računalništvo in informatiko, 2010.

© Razmnoževanje dela v celoti ali po delih je brez  
predhodnega dovoljenja avtorja prepovedano.

---

## 1. Analiza kritične poti

Podan je aciklični neusmerjeni graf akcij (izvedbe projekta). Vozlišča predstavljajo mejnike, povezave pa akcije s pridruženim časom. Naloga je najti kritično (najdaljšo) pot.

Iskanje poti:

a.) rekurzivni eksponentni algoritem (neučinkovit)

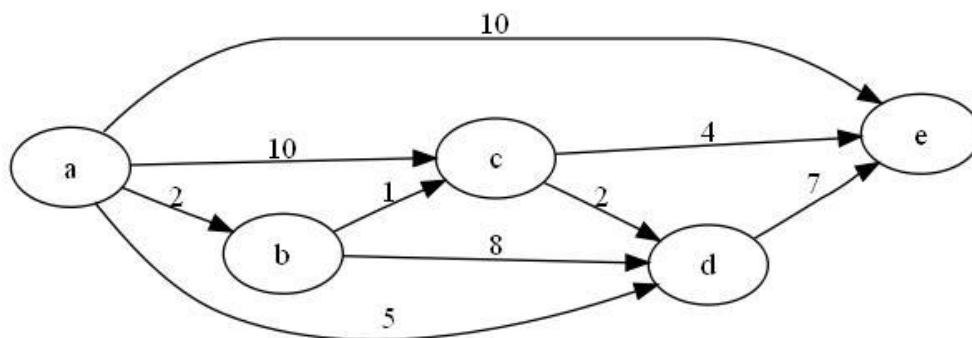
$$t(a,a) = 0$$

$$t(a,c) = \max_{\langle a,b \rangle \in E} [eval(a,b) + t(b,c)]$$

b.) dinamično programiranje: za vsako vozlišče hranimo trojico:

<predhodnik, maksimalni čas do njega, število nepregledanih vhodov>.

Primer:



Vrsta preiskovanih vozlišč: a, b, c, d, e

a: (-, 0, 0)

b: (-, 0, 1), (a, 2, 0)

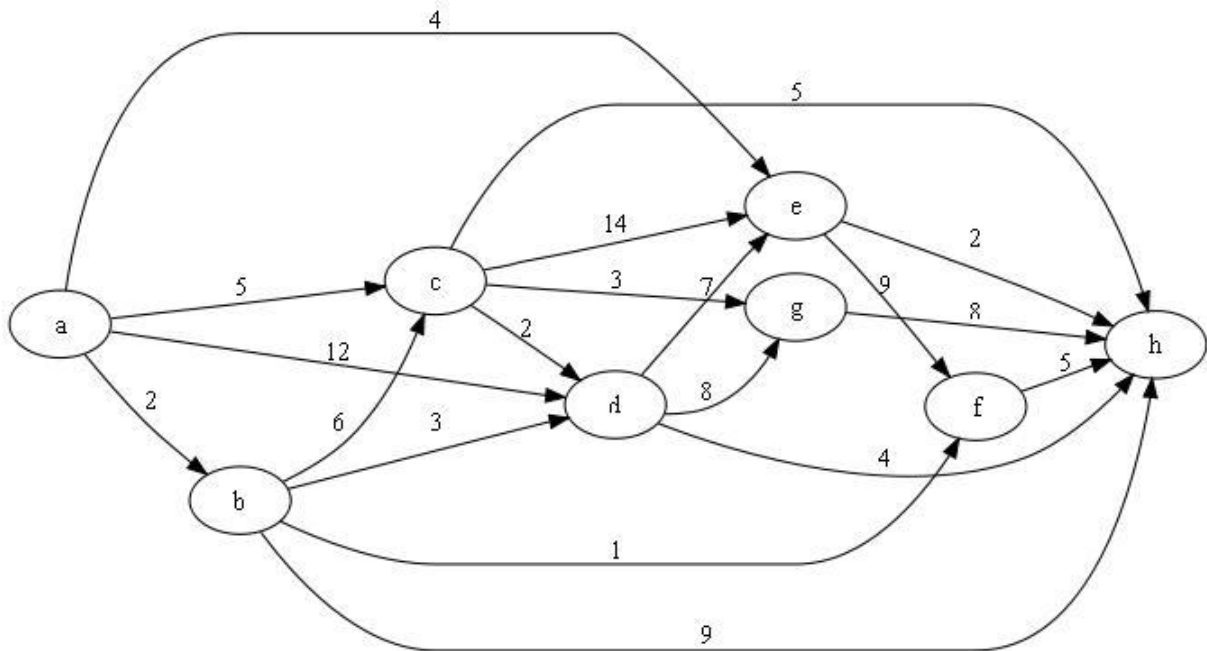
c: (-, 0, 2), (a, 10, 1), (b, 3, 0)

d: (-, 0, 3), (a, 5, 2), (b, 10, 1), (c, 12, 0)

e: (-, 0, 3), (a, 10, 2), (c, 14, 1), (d, 19, 0)

Pot: a-c-d-e v času 19.

Naloga: Poišči kritično pot na naslednjem usmerjenem grafu!



Vrstni red preiskovanja vozlišč: a, b, c, d, e, f, g, h

a: (-, 0, 0)

b:

c:

d:

e:

f:

g:

h:

Rešitev: Pot a-b-c-e-f-h v času 36

## 2. Minimalno vpeto drevo

Cilj: povezati vsa vozlišča v neusmerjenem grafu z minimalnim (najcenejšim) podgrafom.  
Primer uporabe: optimizacije prometnih/komunikacijskih povezav, itd.

Iskanje s požrešnim algoritmom (Primov in Kruskalov).

### Primov algoritem

Vozlišča razdelimo na U in V-U.

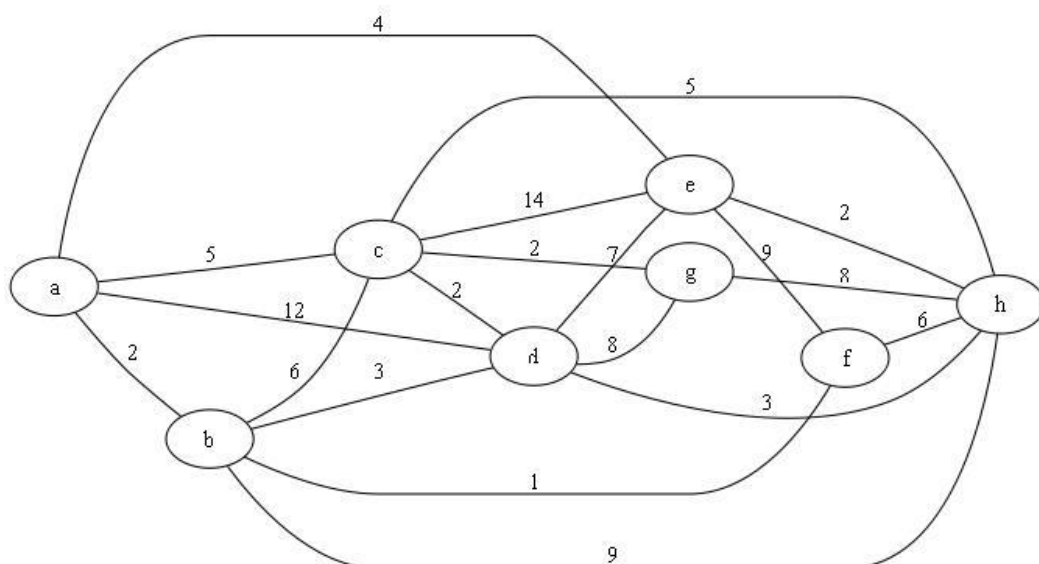
Na začetku je v U samo začetno vozlišče.

Na vsakem koraku dodamo najkrajšo povezavo med U in V-U.

V prioritetni vrsti hranimo evidenco vozlišč iz V-U s ključem, ki predstavlja dolžino najkrajše povezave do tega vozlišča iz U.

Časovna zahtevnost:  $O(n^2)$  ->  $O(|E| \log|V|)$  z uporabo prioritetne vrste.

Primer:



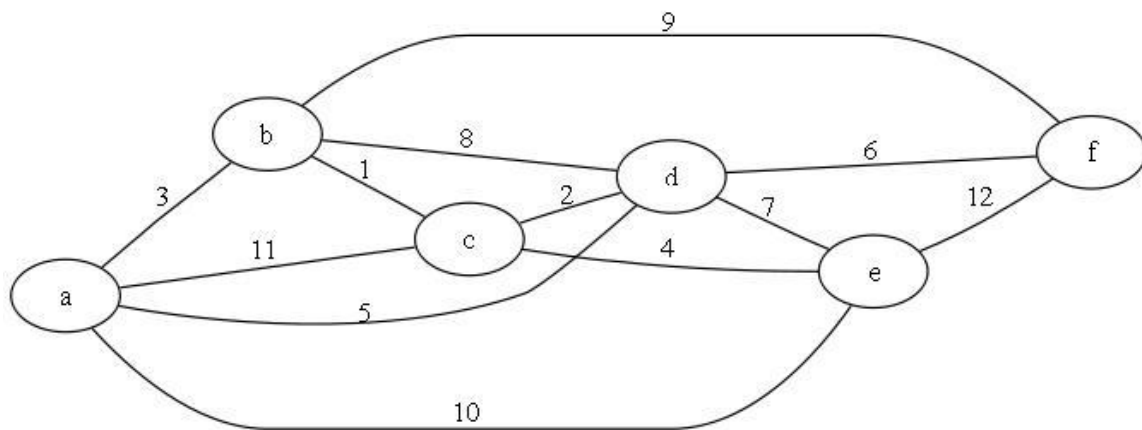
Korak		Prioritetna vrsta
1.		b/2 e/4 c/5 d/12
2.		f/1 d/3 e/4 c/5 h/9
3.		d/3 e/4 c/5 h/6
4.		c/2 h/3 e/4 g/8

<p>5.</p>	<p>Graph diagram for problem 5. Nodes a, b, c, d, f, and h are shaded gray, while e and g are white. Edges and weights: (a,b) weight 2, (b,d) weight 3, (d,f) weight 3, (b,f) weight 1, (c,d) weight 2, (c,g) weight 2, (a,e) weight 4 (dotted), (d,g) weight 2 (dotted), (d,h) weight 3 (dotted), (f,h) weight 3 (dotted).</p>	<p>g/2 h/3 e/4</p>
<p>6.</p>	<p>Graph diagram for problem 6. Nodes a, b, c, d, f, and h are shaded gray, while e and g are white. Edges and weights: (a,b) weight 2, (b,d) weight 3, (d,f) weight 3, (b,f) weight 1, (c,d) weight 2, (c,g) weight 2, (a,e) weight 4 (dotted), (d,g) weight 2 (dotted), (d,h) weight 3 (dotted), (f,h) weight 3 (dotted).</p>	<p>h/3 e/4</p>
<p>7.</p>	<p>Graph diagram for problem 7. Nodes a, b, c, d, f, and h are shaded gray, while e and g are white. Edges and weights: (a,b) weight 2, (b,d) weight 3, (d,f) weight 3, (b,f) weight 1, (c,d) weight 2, (c,g) weight 2, (e,h) weight 2 (dotted), (d,g) weight 2 (dotted), (d,h) weight 3 (dotted), (f,h) weight 3 (dotted).</p>	<p>e/2</p>
<p>8.</p>	<p>Graph diagram for problem 8. Nodes a, b, c, d, f, and h are shaded gray, while e and g are white. Edges and weights: (a,b) weight 2, (b,d) weight 3, (d,f) weight 3, (b,f) weight 1, (c,d) weight 2, (c,g) weight 2, (e,h) weight 2, (d,g) weight 2, (d,h) weight 3.</p>	

Naloga:

Na naslednjem grafu:

- simuliraj Primov algoritem,
- izriši zaporedje kopic, ki jih uporablja algoritem.





### Kruskalov algoritem

Gradimo minimalni vpeti gozd.

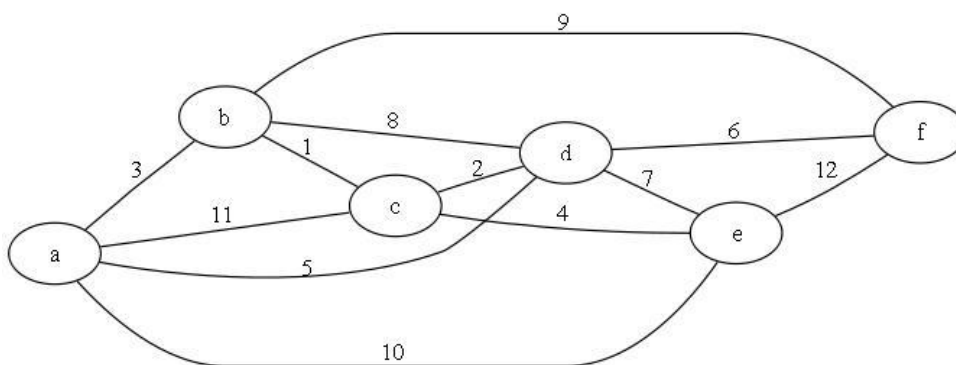
Na začetku je vsako vozlišče eno drevo.

Na vsakem koraku združimo dve drevesi v eno z dodajanjem minimalne povezave.

Prioritetna vrsta hrani vse neuporabljene povezave po ključu – ceni povezave.

Časovna zahtevnost algoritma je  $O(|E|\log|E|)$  pri uporabi prioritetne vrste.

Primer:



Korak		Prioritetna vrsta
1.		b-c/1 c-d/2 a-b/3 c-e/4 a-d/5 d-f/6 d-e/7 b-d/8 b-f/9 a-e/10 a-c/11 e-f/12
2.		c-d/2 a-b/3 c-e/4 a-d/5 d-f/6 d-e/7 b-d/8 b-f/9 a-e/10 a-c/11 e-f/12

3.		$a-b/3$ $c-e/4$ $a-d/5$ $d-f/6$ $d-e/7$ $b-d/8$ $b-f/9$ $a-e/10$ $a-c/11$ $e-f/12$
4.		$c-e/4$ $a-d/5$ $d-f/6$ $d-e/7$ $b-d/8$ $b-f/9$ $a-e/10$ $a-c/11$ $e-f/12$
5.		$a-d/5$ $d-f/6$ $d-e/7$ $b-d/8$ $b-f/9$ $a-e/10$ $a-c/11$ $e-f/12$
6.		$d-e/7$ $b-d/8$ $b-f/9$ $a-e/10$ $a-c/11$ $e-f/12$
7.		

Naloga:

Na naslednjem grafu:

- simuliraj Primov algoritem,
- izriši zaporedje kopic, ki jih uporablja algoritem

