Ekspertski sistemi
Vežbe

Algoritmi pretraživanja

predavač: Dražen Drašković

mart 2015.

Program vežbi

- Algoritmi pretraživanja (+ igre)
- Formalna logika (metodologija predstavljanja znanja)
- Semantičke mreže i okviri?
- Produkcioni sistemi
- Rad u neizvesnom okruženju
- Strategije rešavanja problema

Pretraživanje u Ekspertskim sistemima

- Nalaženje rešenja problema!
- Na primer putovanje iz mesta A u mesto B

Pretraživanje u Ekspertskim sistemima

Putovanje iz mesta A u mesto B

Definisanje problema pretraživanja

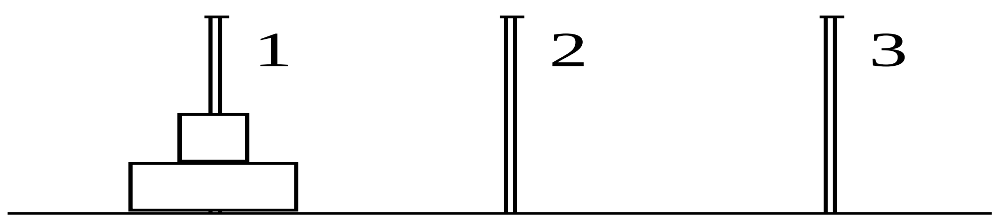
- Stanjima
- Operatorima
- Strategijama pretraživanja
- Funkcijama procene
- Faktorom grananja

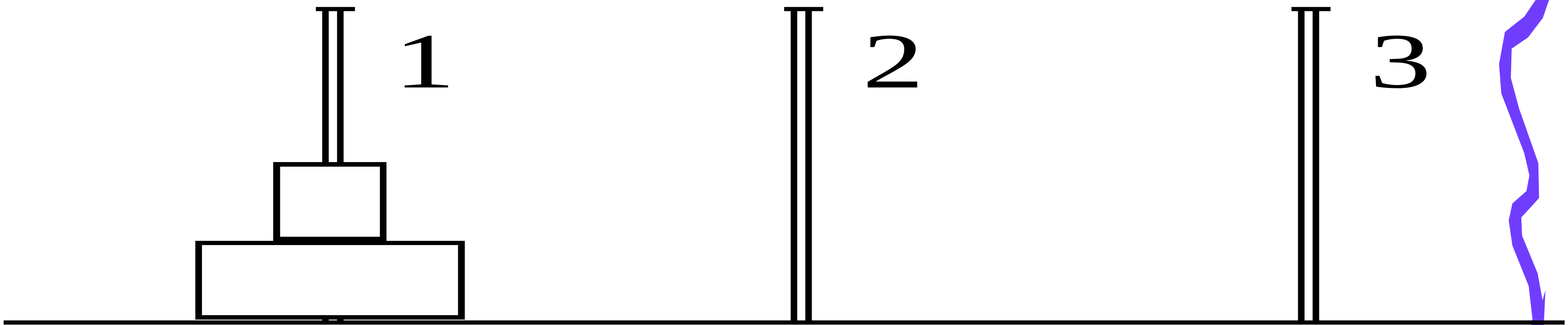
- Iskustvo za prepoznavanje

Zadatak 1: Hanojske kule

Posmatrajmo igru Hanojskih kula, sa dva diska različitih poluprečnika i tri stuba. Cilj igre je da se oba diska sa stuba 1 prebace na stub 3, poštujući sledeća ograničenja:

- U datom trenutku može se pomeriti samo jedan disk
- Veći disk ni u jednom trenutku ne sme da se nadje iznad manjeg




1
2
3

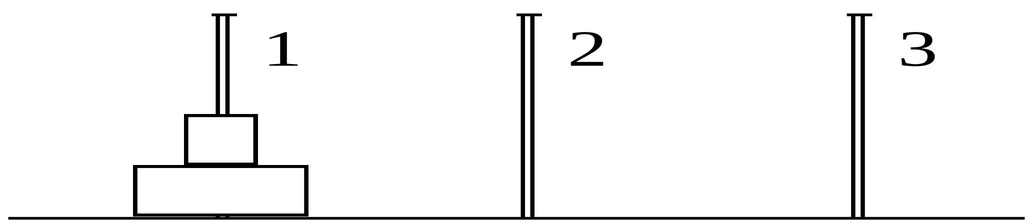
Ako sa (x, y) označimo stanje problem, pri čemu je **x** broj stuba na kome se nalazi veći disk a **y** broj stuba na kome se nalazi manji disk, potrebno je:

a) odrediti dozvoljena stanja

b) formirati tabelu dozvoljenih prelaza izmedju stanja u jednom koraku

c) prikazati kompletan graf pretrage za navedeni problem

d) prikazati kompletno stablo pretrage



a) (odrediti dozvoljena stanja)

Dozvoljena stanja su sva ona stanja (x, y) kod kojih je x, y € {1, 2, 3}. To su stanja:

(1,1) (1,2) (1,3) (2,1) (2,2) (2,3) (3,1) (3,2) (3,3)

Podrazumeva se u slučaju (x,x) da se manji disk nalazi na većem. To ograničenje mora se uzeti u obzir pri definisanju prelaza iz stanja u stanje

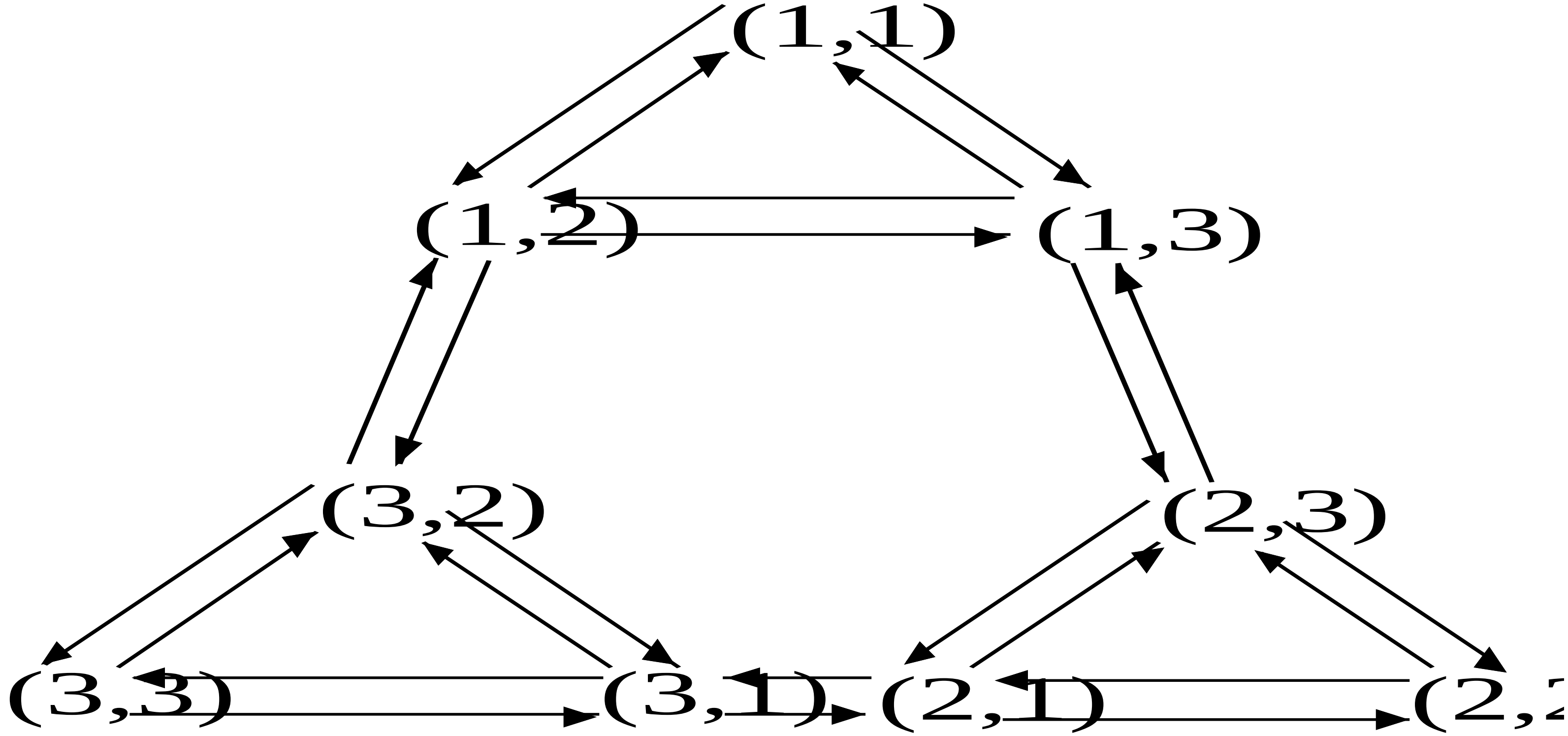
b) (formirati tabelu dozvoljenih prelaza izmedju stanja u jednom koraku)

- Iz stanja (1,1) moguće je pomeriti samo manji disk i to na stub 2 ili na stub 3 što odgovara prelazima u stanja (1,2) odnosno (1,3)
- Iz stanja (1,2) moguće je pomeriti manji disk na stubove 1 ili 3 što odgovara stanjima (1,1) tj. (1,3). Veći disk je moguće pomeriti na stub 3 što odgovara stanju (3,2). Veći disk na stub 2 ??

	(1,1)	(1,2)	(1,3)	(2,1)	(2,2)	(2,3)	(3,1)	(3,2)	(3,3)
(1,1)		X	X						
(1,2)	X		X					X	
(1,3)	X	X				X			
(2,1)					X	X	X		
(2,2)				X		X			
(2,3)			X	X	X				
(3,1)				X				X	X
(3,2)		X					X		X
(3,3)							X	X	

c) (prikazati kompletan graf pretrage za nevedeni problem)

Graf se dobija na osnovu prethodne tabele

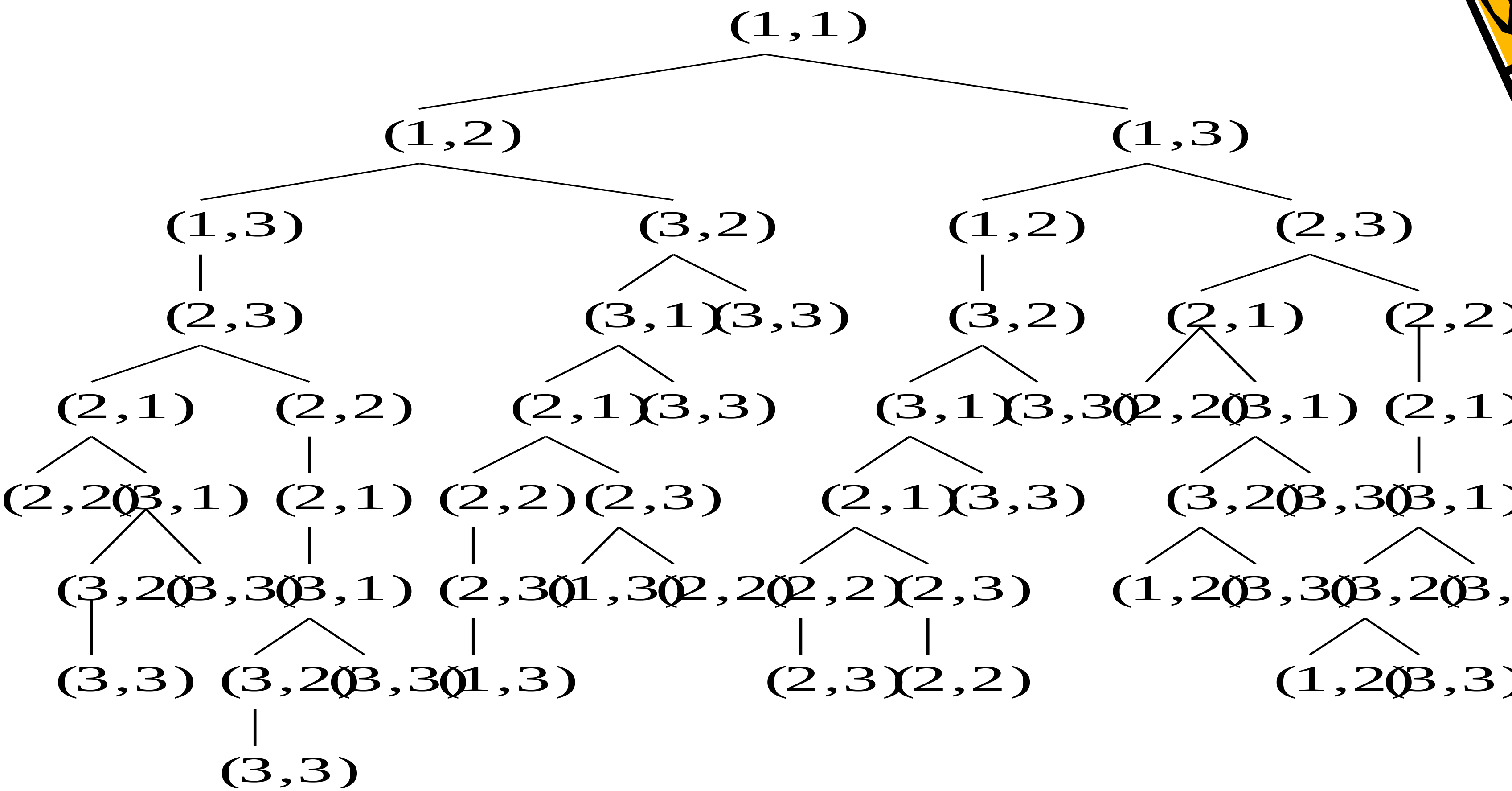


d) (prikazati kompletno stablo pretrage)

Kompletno stablo pretrage obuhvata sve otvorene putanje u grafu pretrage koje počinju u startnom čvoru i završavaju se ili u ciljnom čvoru ili u čvoru iz koga svaka dalja primena operatora dovodi do zatvaranja putanje

- Ukoliko cilj nije eksplicitno zadat, pretraga se završava kada se obiđu sva stanja iz grafa pretrage

- Zatvorena putanja - ako se na putanji dva puta pojavljuje isti čvor
- Otvorena putanja u suprotnom
- Konstrukcija stabla pretrage:
  - startnom stanju odgovara koren stabla pretrage
  - vrši se ekspandovanje startnog čvora
  - u stablo se unose njegovi sinovi i grane
  - bira se jedan od neobradjenih čvorova. Ukoliko je ciljni čvor onda je kraj, u suprotnom se ekspanduje stanje koje odgovara tom čvoru
  - sve dok postoje neobradjeni čvorovi


(1,1)
(1,2) (1,3)
(1,3) (3,2) (1,2) (2,3)
(2,3) (3,1) (3,3) (3,2) (2,1) (2,2)
(2,1) (2,2) (2,1) (3,3) (3,1) (3,3) (2,2) (3,1) (2,1)
(2,2) (3,1) (2,1) (2,2) (2,3) (2,1) (3,3) (3,2) (3,3) (3,1)
(3,2) (3,3) (3,1) (2,3) (1,3) (2,2) (2,2) (2,3) (1,2) (3,3) (3,2)
(3,3) (3,2) (3,3) (1,3) (2,3) (2,2) (1,2) (3,3)
(3,3)

Dodatno rešiti

- Rešiti problem sa tri diska
- Postavka je identična

Zadatak 2: Problem dva krčaga

Na raspolaganju su dva krčaga zapremina 3 i 2 litra bez mernih oznaka. Svaki krčag može da se puni vodom sa česme, a voda može i da se prospe. Potrebno je postići da se u manjem krčagu nađe 1 litar vode

3l

2l

a) Definisati prostor stanja problema
b) Definisati operatore koji sistem prevode iz stanja u stanje
c) Navesti jedan od redosleda primene operatora koji predstavlja rešenje problema
d) Tabelarno predstaviti kompletan graf pretraživanja za dati problem

a) (Definisati prostor stanja problema)

Stanja: uredjeni par (x,y)

- **x** količina vode u većem krčagu
- **y** količina vode u manjem krčagu
- Ograničenja: 0 <= x <= 3, 0 <= y <=2
- (0,0) startno stanje
- (x,1) ciljno stanje (+ ⬆ ograničenja)

b) (Definisati operatore koji sistem prevode iz stanja u stanje)

- Napomena: jedno od mogućih rešenja
- ideje ??

Ideje:

- Isprazniti krčag
- Napuniti krčag
- Izvršiti presipanje

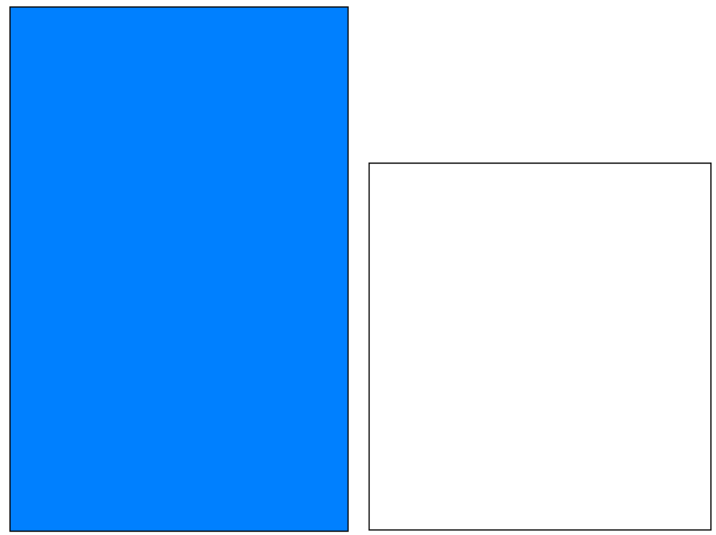
- Usvajamo sledećih osam operatora:

Redni broj	akcija	Tekuće stanje	novo stanje	uslov primene
1.	isprazni veći krčag	(x, y)	$(0,y)$	$x > 0$
2.	isprazni manji krčag	(x, y)	$(x,0)$	$y > 0$
3.	napuni veći krčag iz česme	(x, y)	$(3,y)$	$x < 3$
4.	napuni manji krčag iz česme	(x, y)	$(x,2)$	$y < 2$
5.	napuni veći krčag iz manjeg	(x, y)	$(3,y\text{-}3+x)$	$x < 3$ i $y > 0$ i $x+y>=3$
6.	napuni manji krčag iz većeg	(x, y)	$(x\text{-}2+y,2)$	$x > 0$ i $y < 2$ i $x+y>=2$
7.	isprazni veći krčag u manji	(x, y)	$(0,x+y)$	$x > 0$ i $y < 2$ i $x+y<=2$
8.	isprazni manji krčag u veći	(x, y)	$(x+y,0)$	$x < 3$ i $y > 0$ i $x+y<=3$

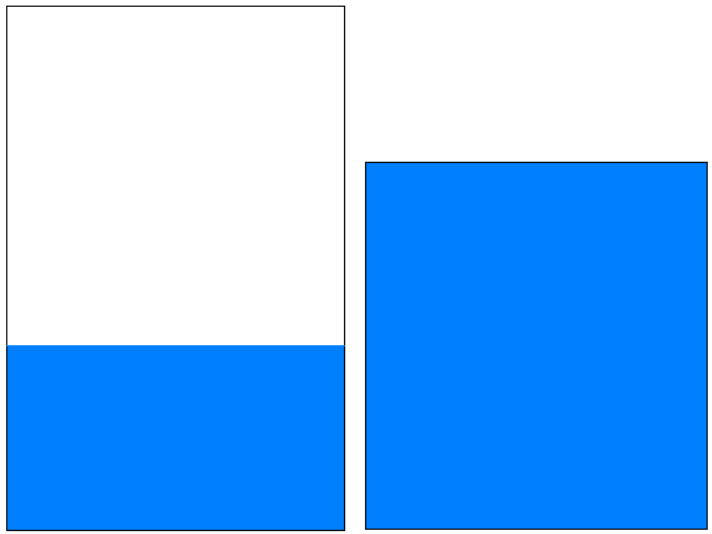
c) (Navesti jedan od redosleda primene operatora koji predstavlja rešenje problema)

- 3,6,1,8,4,5

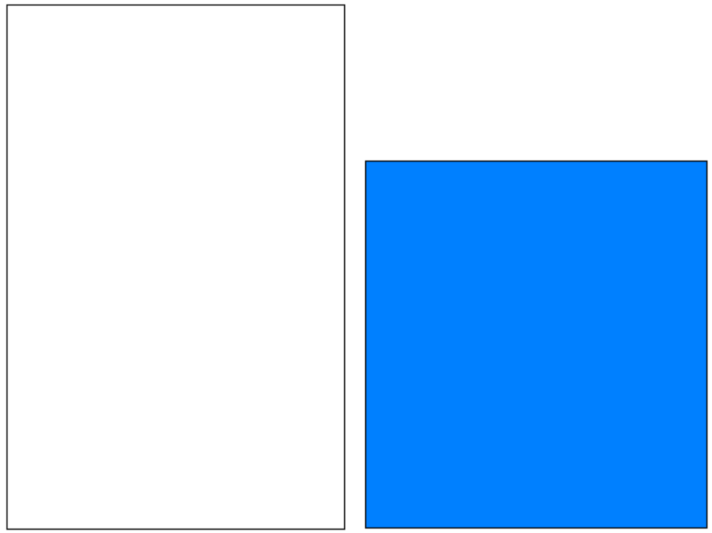
- <u>3</u>,6,1,8,4,5 (napuni veći krčag iz česme)



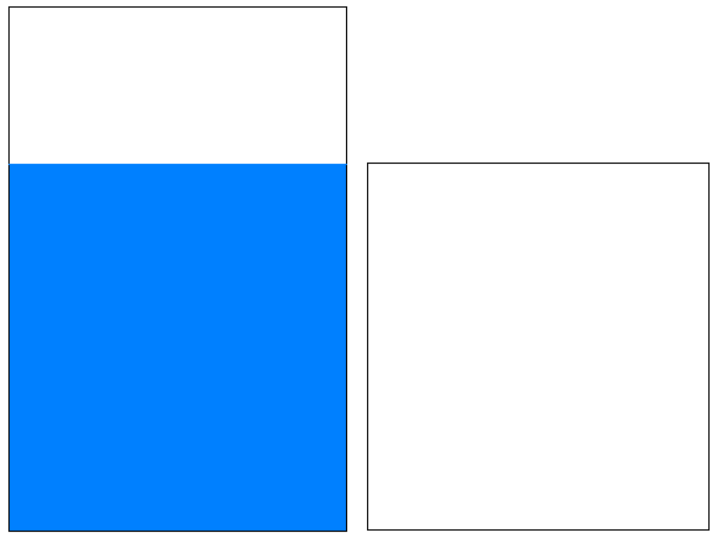
- 3,<u>6</u>,1,8,4,5 (napuni manji krčag iz većeg)



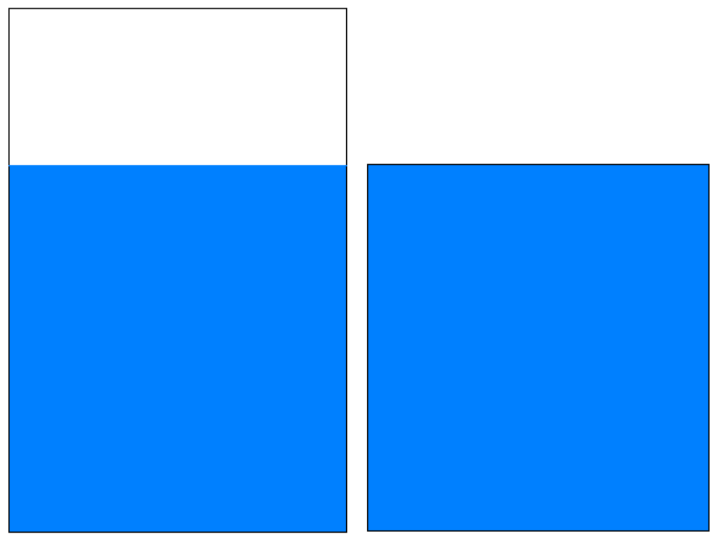
- 3,6,1,8,4,5 (isprazni veći krčag)



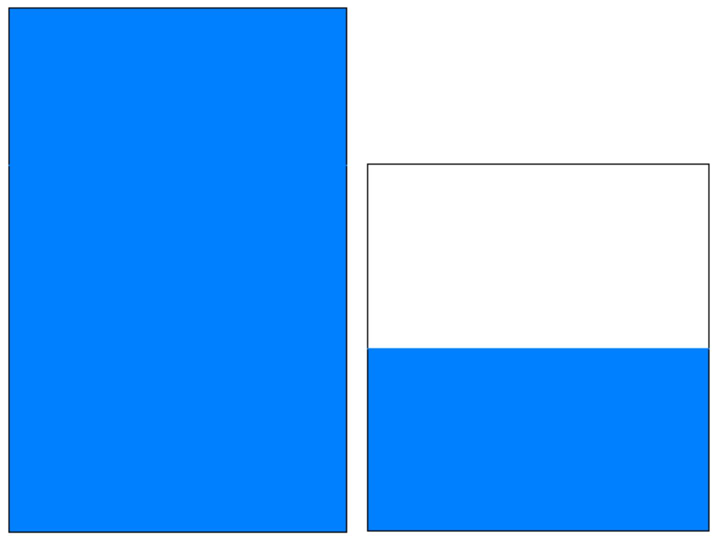
- 3,6,1,8,4,5 (isprazni manji krčag u veći)



- 3,6,1,8,4,5 (napuni manji krčag sa česme)



- 3,6,1,8,4,5 (napuni veći krčag iz manjeg)



d) (Tabelarno predstaviti kompletan graf pretraživanja za dati problem)

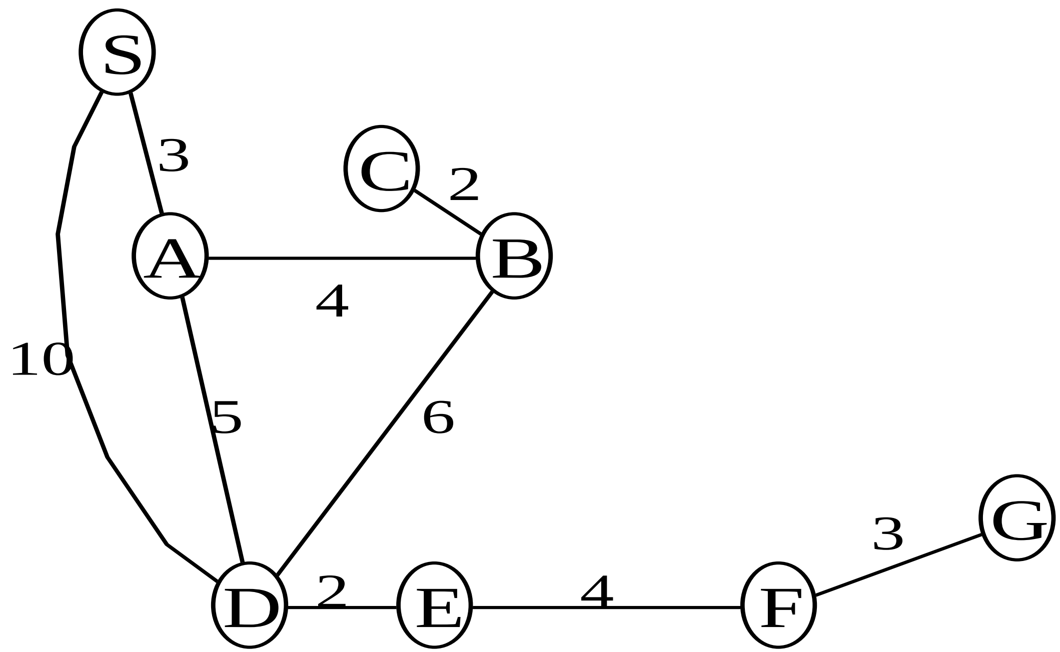
- Iz kog stanja pod dejstvom kog operatora možemo preći u koje stanje
- OpB gde je B broj operatora
- Prazan ulaz u tabeli prikazuje nemoguće prelaze

	(0,0)	(3,0)	(0,2)	(3,2)	(1,2)	(2,0)	(1,0)	(2,2)	(0,1)	(3,1)
(0,0)		op3	op4							
(3,0)	op1			op4	op6					
(0,2)	op2			op3		op8				
(3,2)		op2	op1							
(1,2)		op5,8	op1	op3			op2			
(2,0)	op1	op3	op6,7					op4		
(1,0)	op1	op3			op4				op7	
(0,1)	op2		op4				op8			op3
(3,1)		op2		op4				op6	op1	

- Prelaz iz (1,2) u (3,0)
- Prelaz iz (2,0) u (0,2)
- Granični slučajevi!

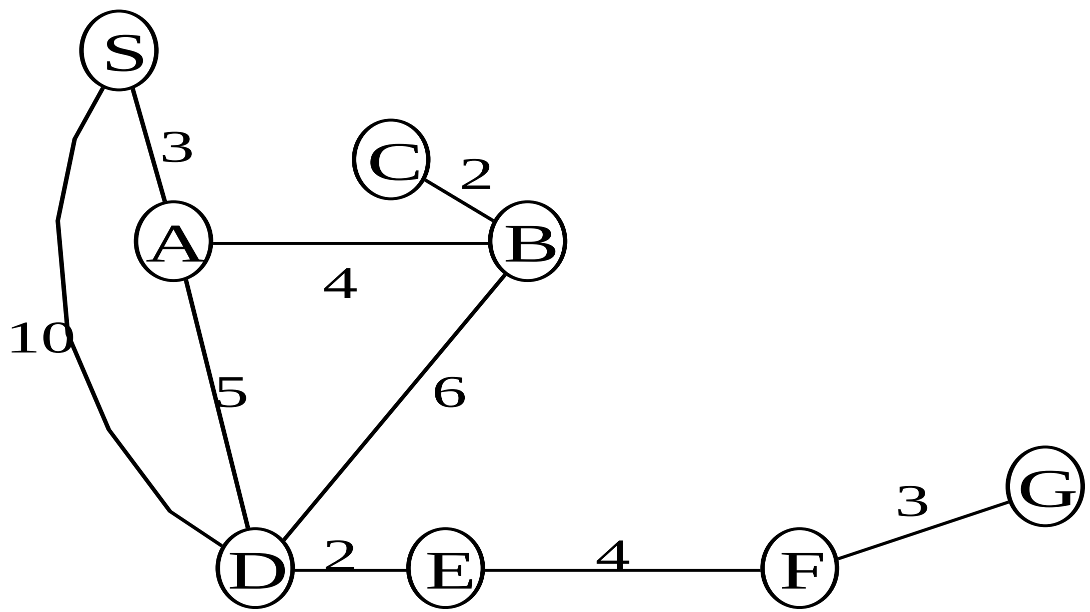
Zadatak 3: Putna mreža (razni algoritmi)

Na slici je prikazana mreža puteva sa označenim dužinama puteva u kilometrima



Vazdušna rastojanja od pojedinih gradova do grada G u kilometrima data su tabelom.

Grad	S	A	B	C	D	E	F
Rastojanje do G	11.5	10.4	6.7	7.0	8.9	6.9	3.0



Prikazati stablo pretrage i navesti redosled obilaženja čvorova pri pretrazi za nalaženje puta između gradova S i G ako se koristi :

a) pretraga po dubini (*depth-first*)

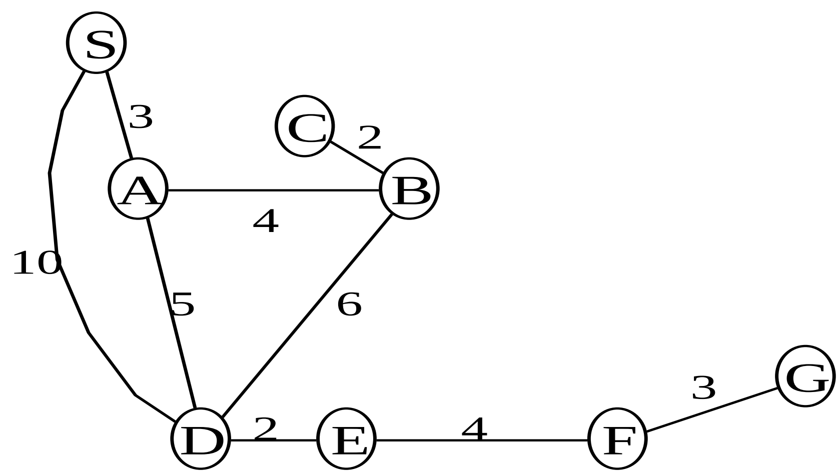
b) pretraga po širini (*breadth-first*)

c) planinarenje (*hill-climbing*)

d) prvo najbolji (*best first*)

e) grananje i graničavanje (*branch and bound*)

f) A*



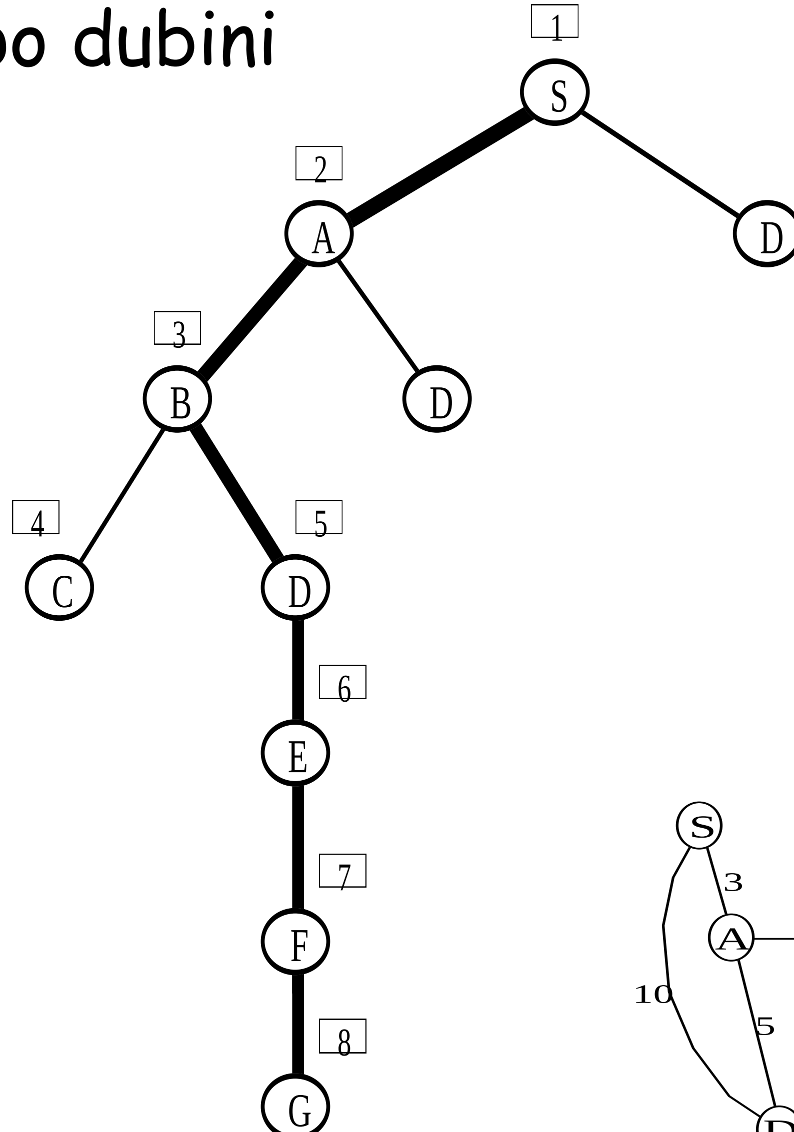
- Važno: definisati na pogodan način heurističku funkciju i cenu rešenja za metode kojima su ove veličine potrebne

- Za koje algoritme je potrebno definisati prethodne veličine ?

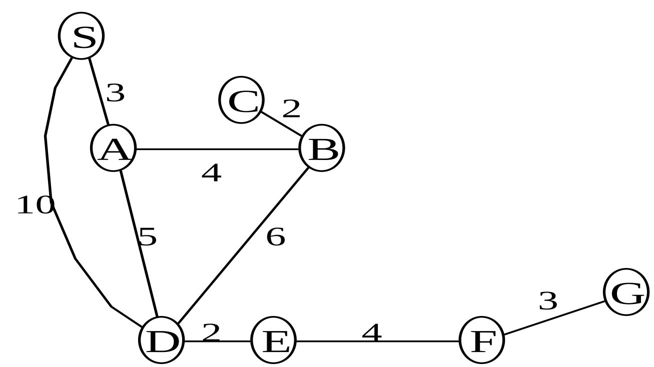
a) (Pretraga po dubini)

- Lista čvorova sadrži startni čvor
- Uklanja se čvor sa početka liste
  - Ako je u pitanju ciljni čvor završiti pretragu
  - Ako nije ciljni dodati njegove sledbenike na početak liste (ukoliko postoje). Ponoviti ovaj korak.
- Ako je ciljni čvor pronadjen pretraga je uspešna

Pretraga po dubini



Dužina = 22

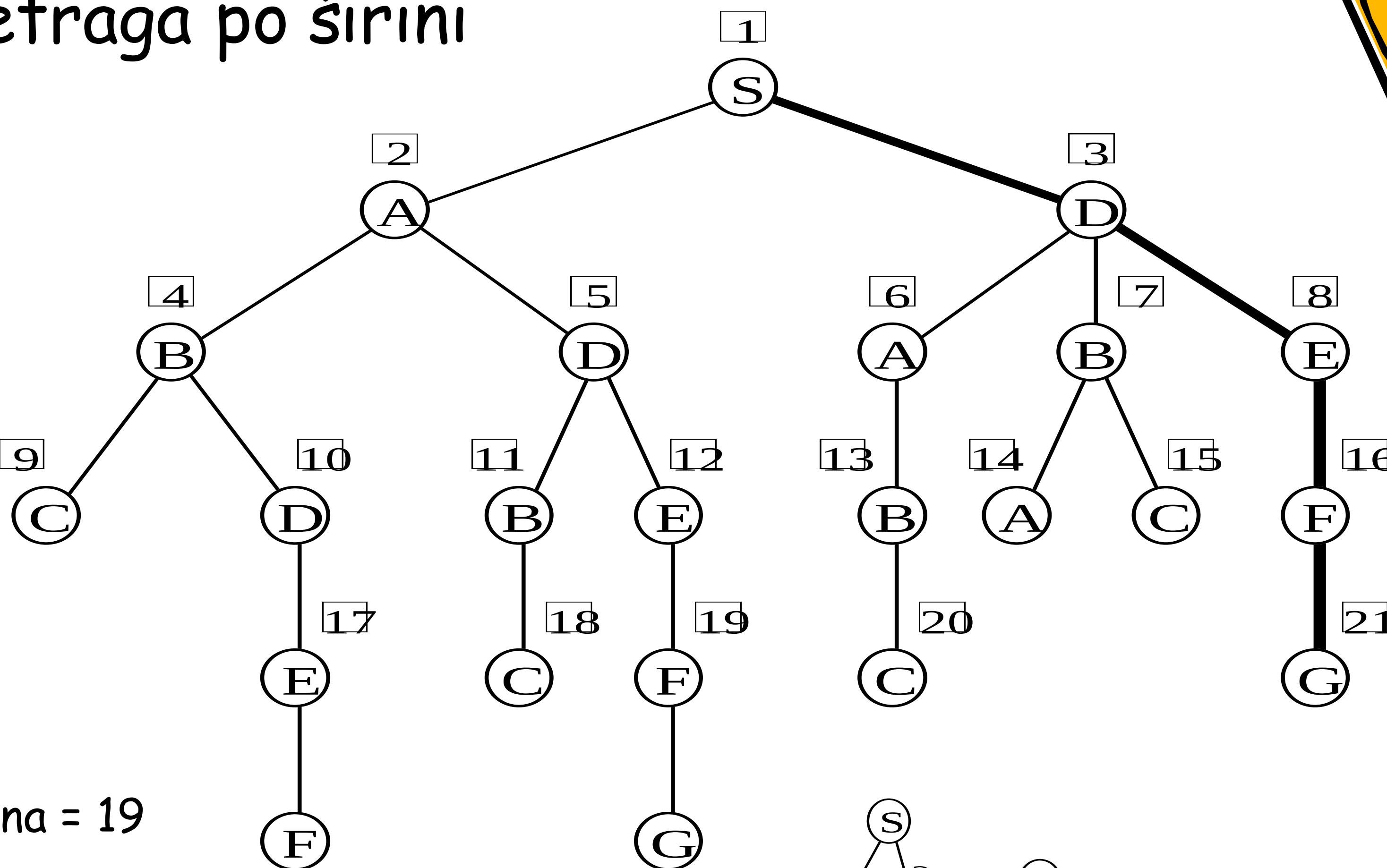


- Da li je pronadjena putanja najkraća moguća putanja ?

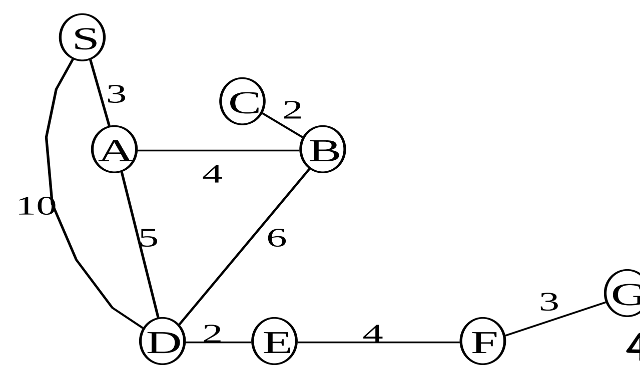
b) (Pretraga po širini)

- Lista čvorova sadrži startni čvor
- Uklanja se čvor sa početka liste
  - Ako je u pitanju ciljni čvor završiti pretragu
  - Ako nije ciljni dodati njegove sledbenike na kraj liste (ukoliko postoje). Ponoviti korak
- Ako je ciljni čvor pronadjen pretraga je uspešna

Pretraga po širini



Dužina = 19



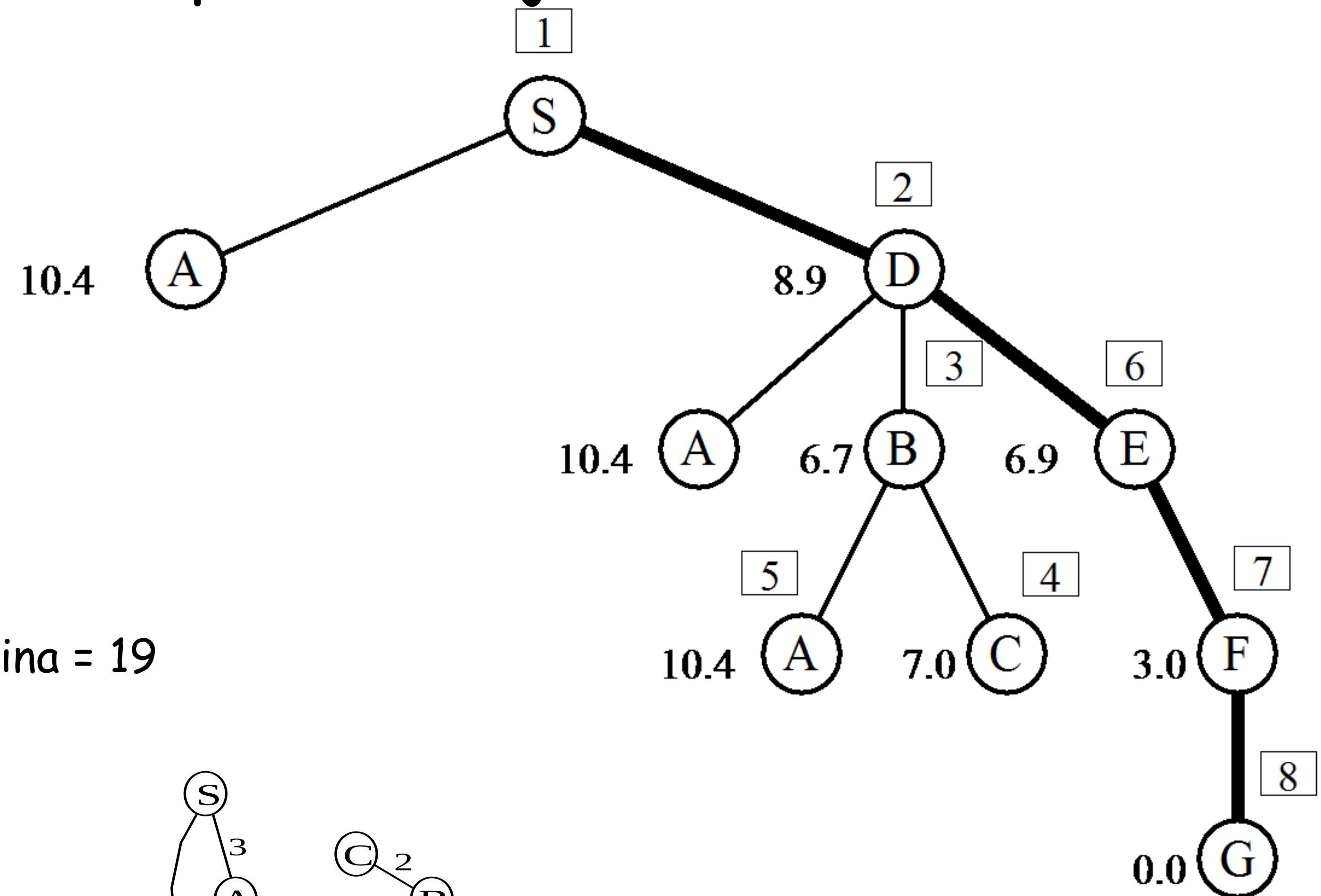
- Mana ?

- Da li je put najkraći mogući ?

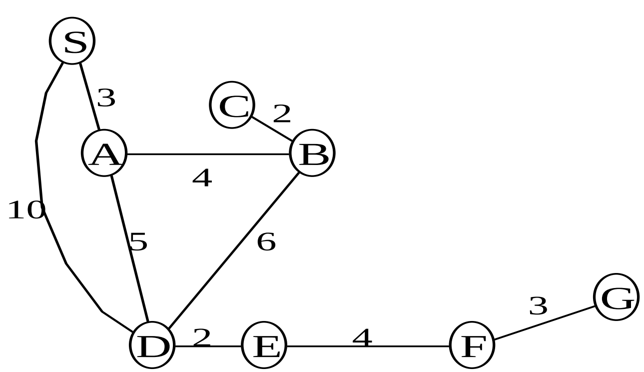
c) (Metoda planinarenja)

- Lista čvorova sadrži startni čvor
- Uklanja se čvor sa početka liste
  - Ako je u pitanju ciljni čvor završiti pretragu
  - Ako nije ciljni, dodati njegove sledbenike na početak liste (ukoliko postoje). Sledbenike treba urediti rastuće na osvnovu vrednosti heurističke funkcije. Prvi - najmanja vrednost. Ponoviti korak.
- Ako je ciljni čvor pronadjen pretraga je uspešna

Metoda planinarenja



Dužina = 19

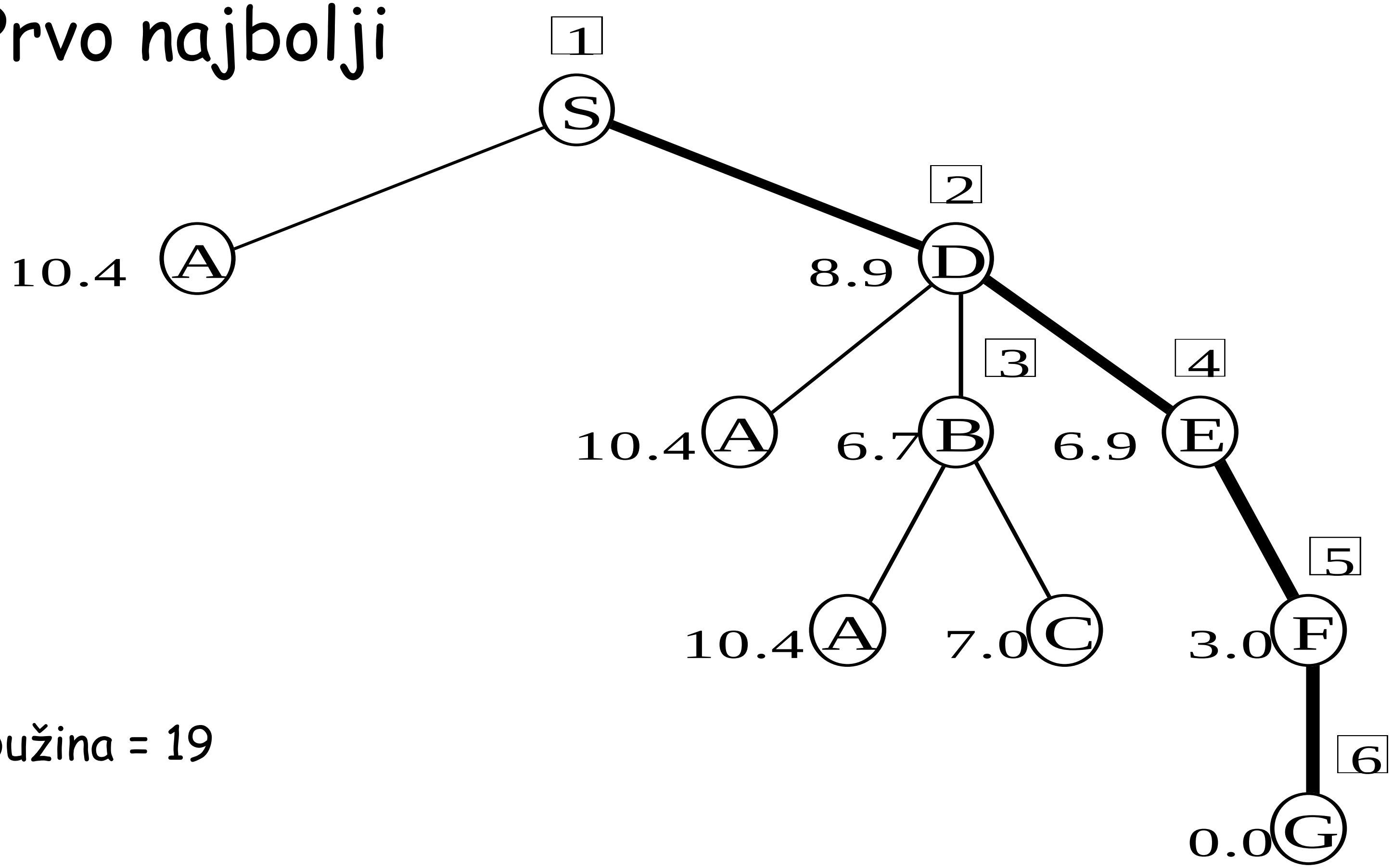


- Nedostatak metode planinarenja ?
- Zašto čvor A nije dalje razvijan ?

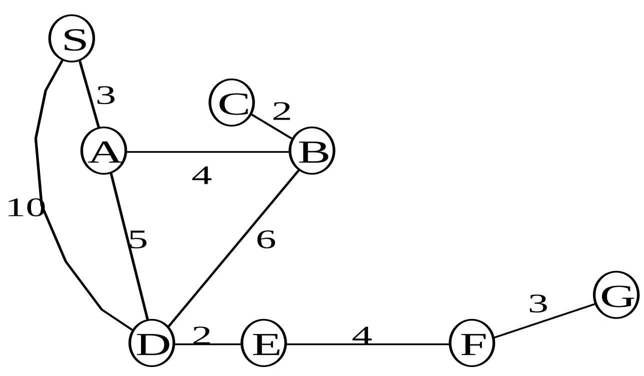
d) (Prvo najbolji)

- Lista čvorova sadrži startni čvor
- Uklanja se čvor sa početka liste
  - Ako je u pitanju ciljni čvor završiti pretragu
  - Ako nije ciljni, dodati njegove sledbenike u listu (ukoliko postoje). <u>Celokupnu listu</u> sortirati rastuće. Ponoviti korak
- Ako je ciljni čvor pronadjen pretraga je uspešna

Prvo najbolji



Dužina = 19

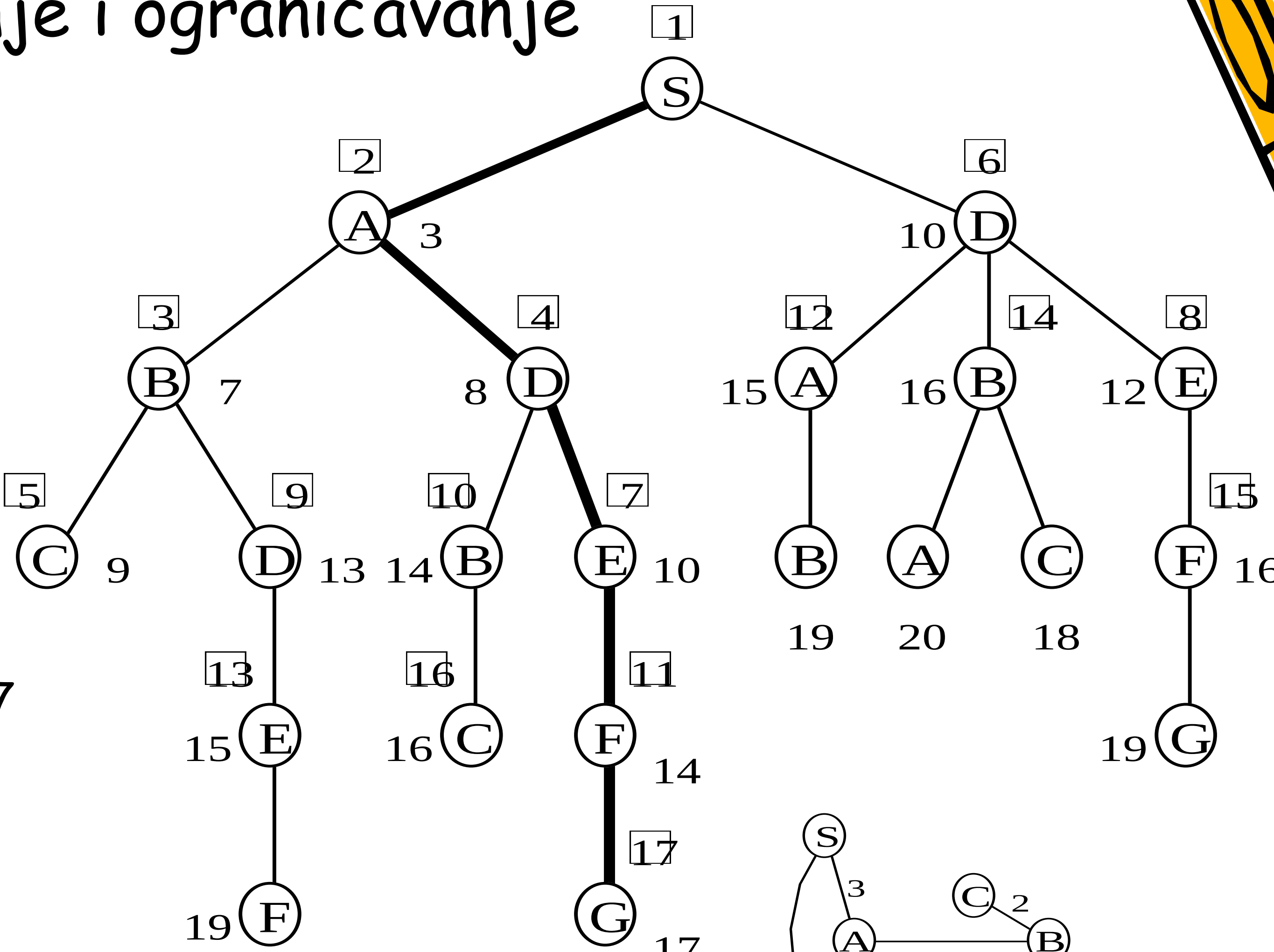


- Šta je prednost ?

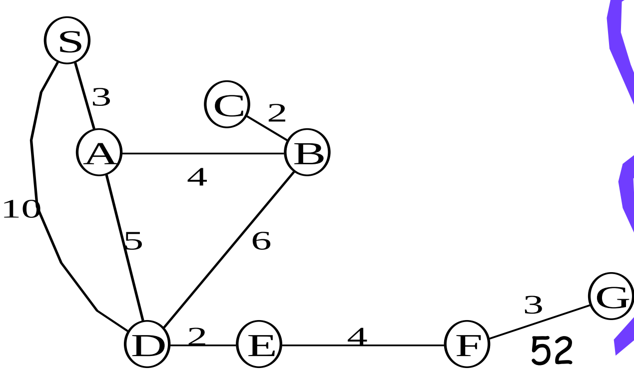
e) (Grananje i ograničavanje)

- Lista parcijalnih putanja sadrži jednu putanju dužine nula (startni čvor)
- Uklanja se putanja sa početka liste
  - Ako putanja dostiže ciljni čvor pretraga se završava
  - Za svaki sledbenik poslednjeg čvora na uklonjenoj putanji formira se po jedna nova putanja.
  - Za svaku putanju izračuna se cena koštanja
  - Nove putanje se ubace u listu koja se zatim sortira rastuće na osnovu cena koštanja
- Ako je ciljni čvor pronadjen pretraga je uspešna

Grananje i ograničavanje



Dužina = 17

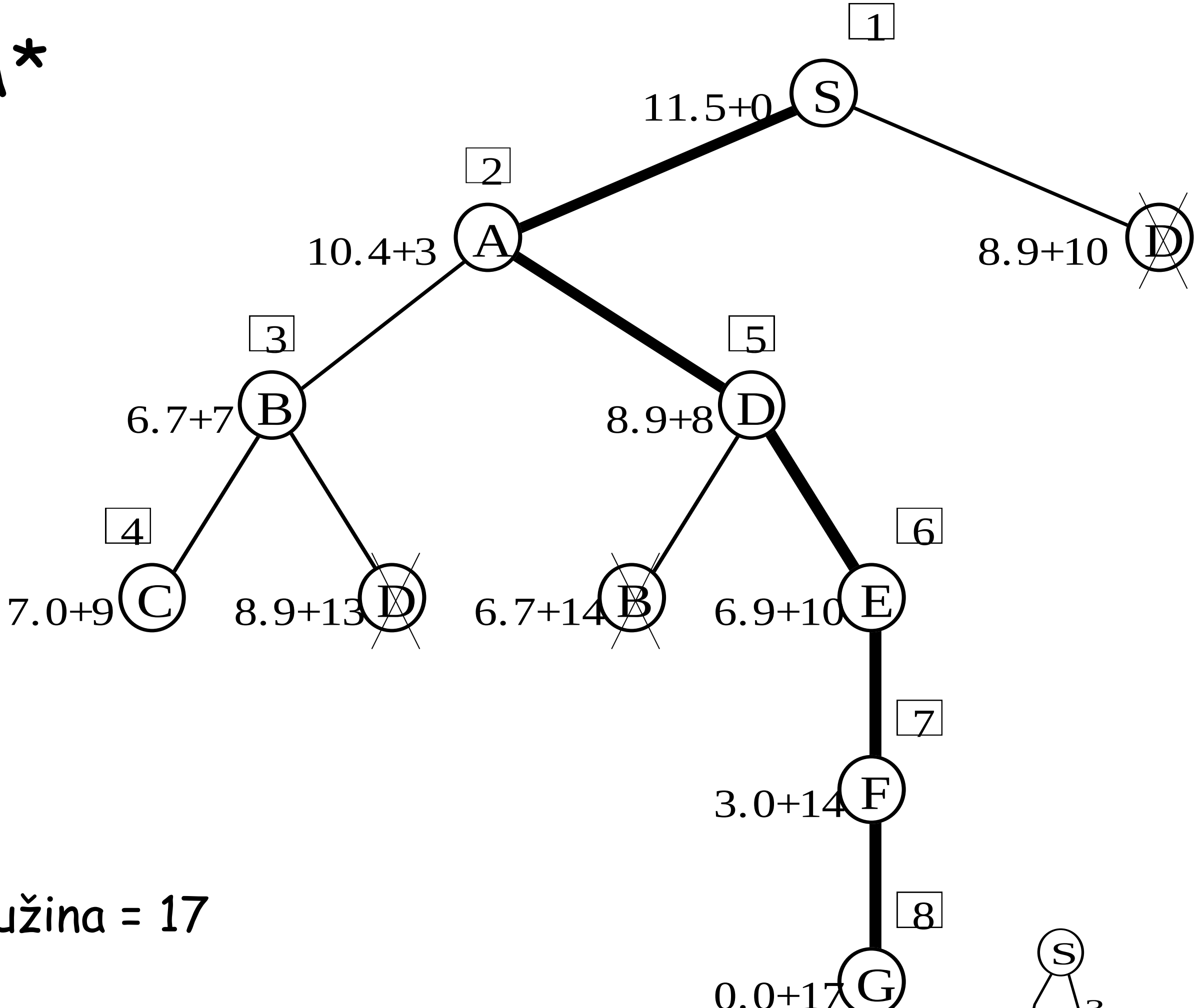


- Pretraga se ne završava kada se ekspanduje čvor F i otkrije čvor G
- Dobijena je optimalna putanja (u kom smislu optimalna?)

f)(A*)

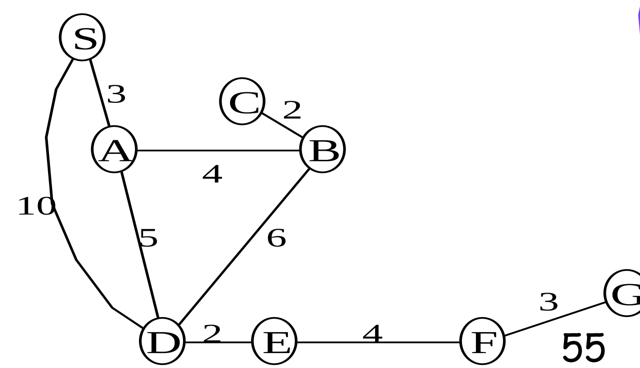
- Lista parcijalnih putanja sadrži jednu putanju dužine nula (startni čvor)
- Uklanja se putanja sa početka liste
  - Ako putanja dostiže ciljni čvor pretraga se završava
  - Za svaki sledbenik poslednjeg čvora na uklonjenoj putanji formira se po jedna nova putanja
  - Za svaku putanju izračuna se <u>kumulativna cena koštanja</u>. Za poslednji čvor na putanji izračunati h. Funkcija procene za svaku putanju je: f = h + c
  - Nove putanje se ubace u listu koja se zatim sortira rastuće na osnovu funcija procene
- Ako je ciljni čvor pronadjen pretraga je uspešna

A*



1
S
11.5+0
2
A
10.4+3
D
8.9+10
3
B
6.7+7
5
D
8.9+8
4
C
7.0+9
D
8.9+13
B
6.7+14
6
E
6.9+10
7
F
3.0+14
8
G
0.0+17

Dužina = 17



S
A
B
C
D
E
F
G
3
2
4
10
5
6
2
4
3

Dinamičko programiranje

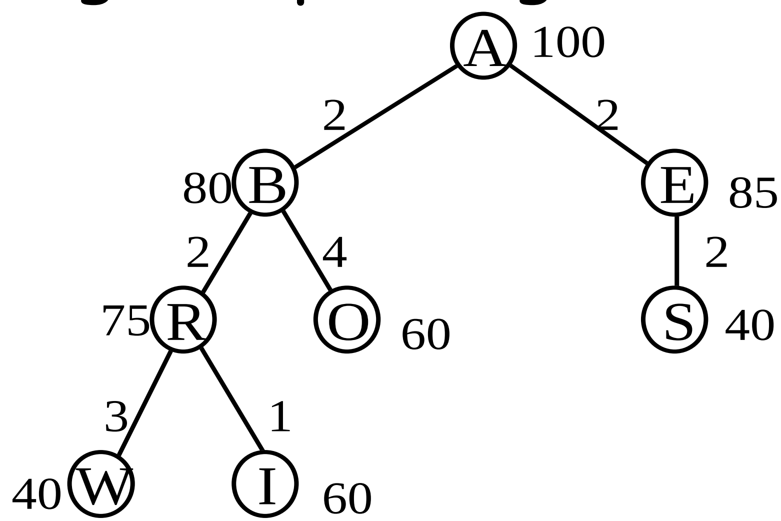
- **Dinamičko programiranje** je metod kojim se smanjuje vreme izvršavanja onih problema u kojima se zahteva traženje optimalne podstrukture i koji imaju potprobleme koji se ponavljaju

Beam pretraga (nalik best first)

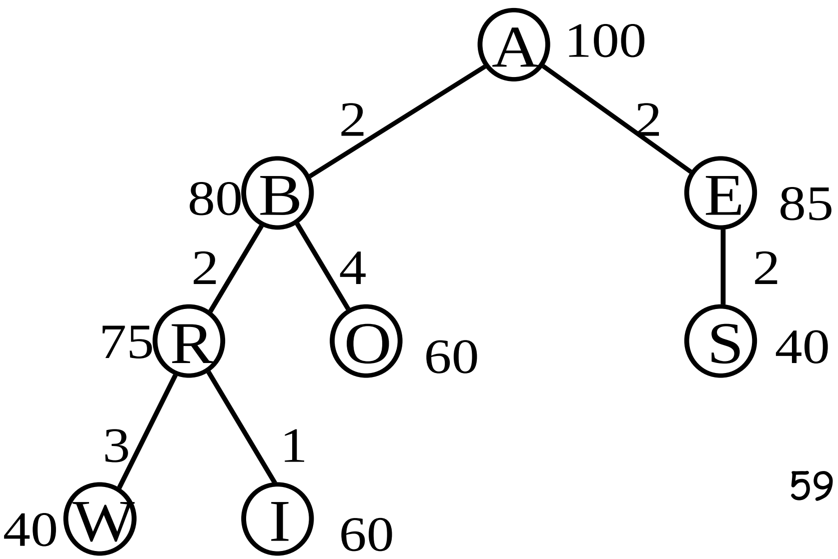
- Lista čvorova sadrži startni čvor
- Uklanja se čvor sa početka liste
  - Ako je u pitanju ciljni čvor završiti pretragu
  - Ako nije ciljni dodati njegove sledbenike u listu (najboljih m). Ponoviti korak
- Ako je ciljni čvor pronadjen pretraga je uspešna

Zadatak 4: Samoglasnici i Suglasnici

Posmatrajmo potpuni *graf* pretrage prikazan na slici. Vrednosti heurističke funkcije prikazane su pored svakog čvora, dok su cene operatora promene stanja prikazane pored grana grafa pretrage

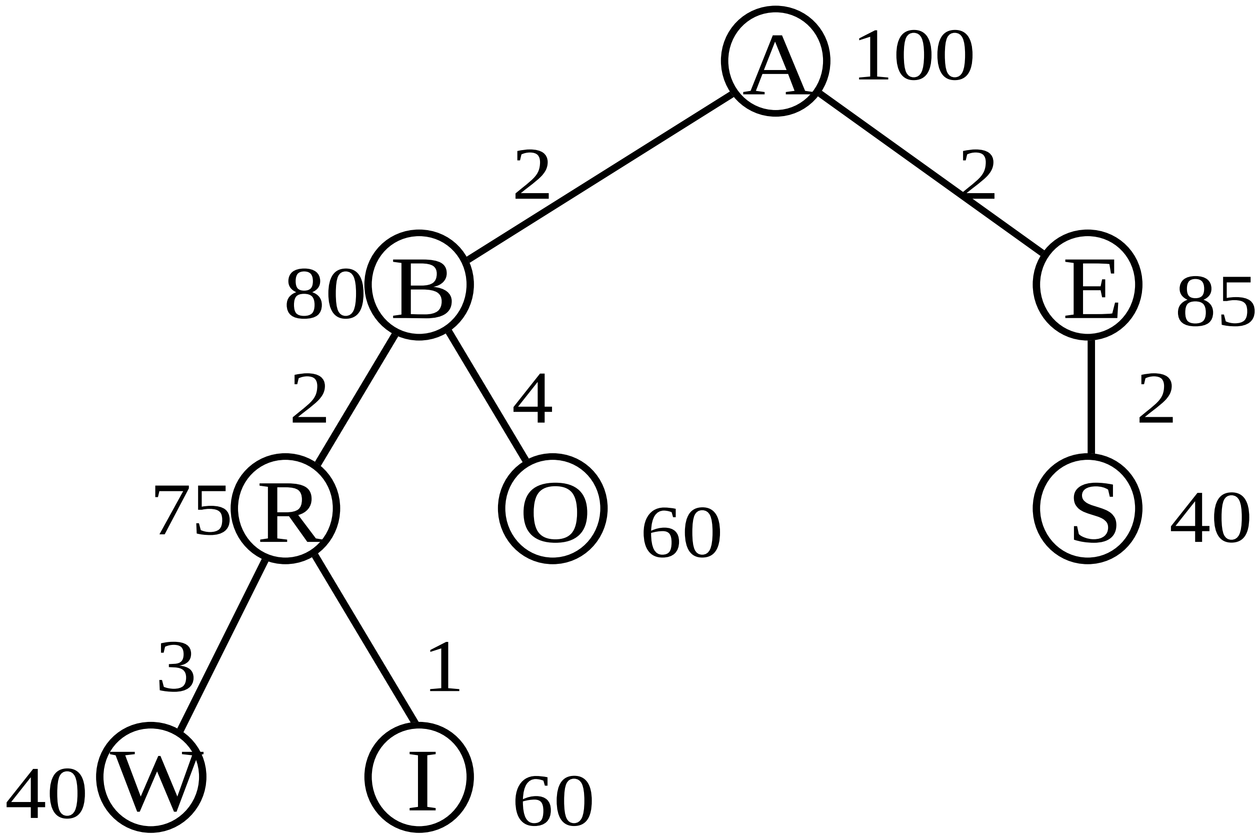


a) Koje je stanje ekspandovano četvrto u pretraživanju metodom po dubini uz primenu heuristike da stanja čija su imena samoglasnici imaju prednost u odnosu na ostala (leksikografski poredak)?
b) Koje je stanje ekspandovano četvrto primenom metode pretraživanja 'prvo najbolji' ?



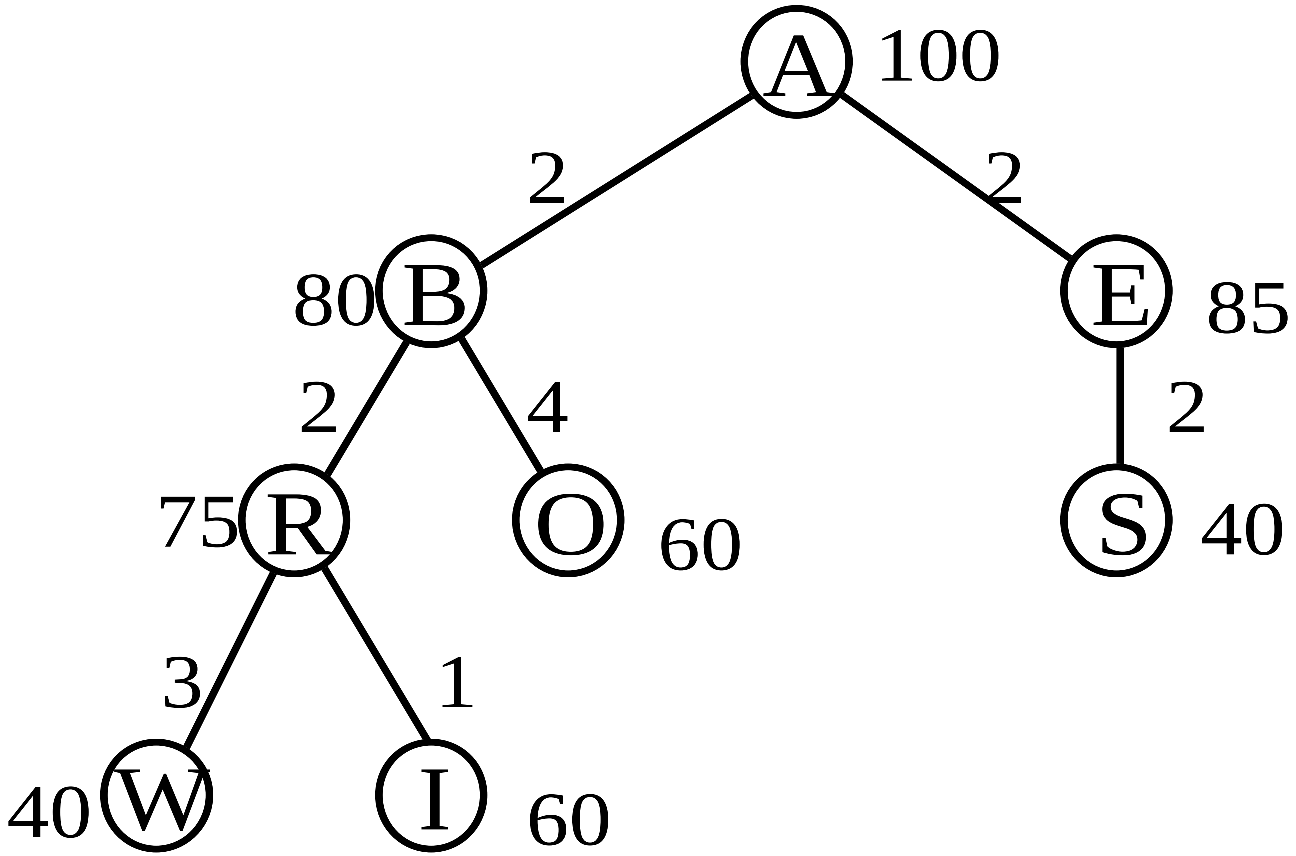
a) (metod po dubini)

- Ne koristi se data heuristička funkcija
- Ne koristi se data funkcija cene koštanja
- Prednost koja je data samoglasnicima odredjuje lokalno najbolji čvor medju sledbenicima tekućeg čvora
- O kom se ustvari pretraživanju radi ?


A 100
2
2
80 B
E 85
2
4
2
75 R
O 60
S 40
3
1
40 W
I 60

b) (prvo najbolji)

- Šta koristimo od datih informacija?
- Pri ovoj pretrazi koristi se definisana heuristička funkcija
- Ne koristi se uvedena funkcija cene koštanja


A 100
2
2
80 B
E 85
2
4
2
75 R
O 60
S 40
3
1
40 W
I 60

Zadatak 5: Tri operatora

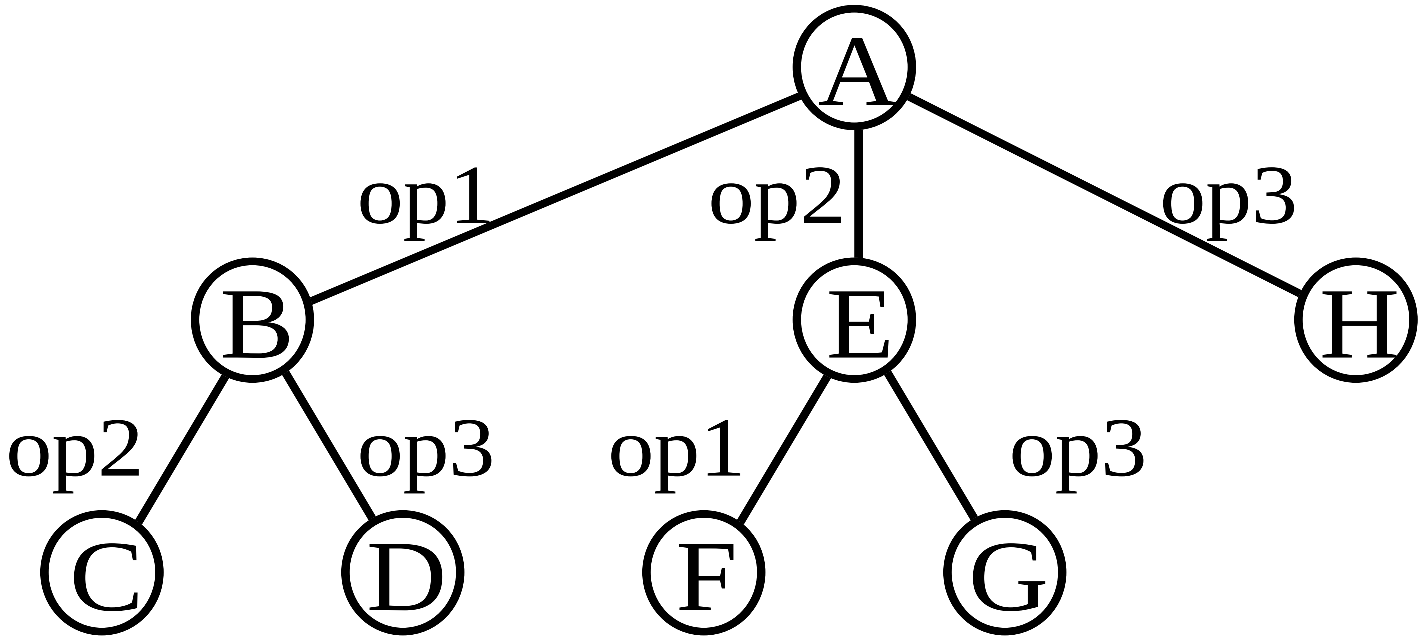
- Poznato je da za neki problem pretraživanja postoje tri operatora, *op1*, *op2* i *op3*
- U polaznom stanju može se primeniti bilo koji od njih
- Ako prvi primenjeni operator nije bio *op3*, tada se u sledećem koraku može primeniti operator različit od prvog
- Svaka dalja primena nekog od operatora nije dozvoljena
- Kako cilj nije dat, pretraživanje mora da obuhvati sva moguća stanja

a) Prikazati graf pretraživanja i označiti čvorove prema redosledu obilaženja, predpostavljajući da *op1* ima prednost nad *op2*, a ovaj takođe ima prednost nad *op3*. Koristiti strategiju pretraživanja po dubini. Stanja obeležiti simbolički

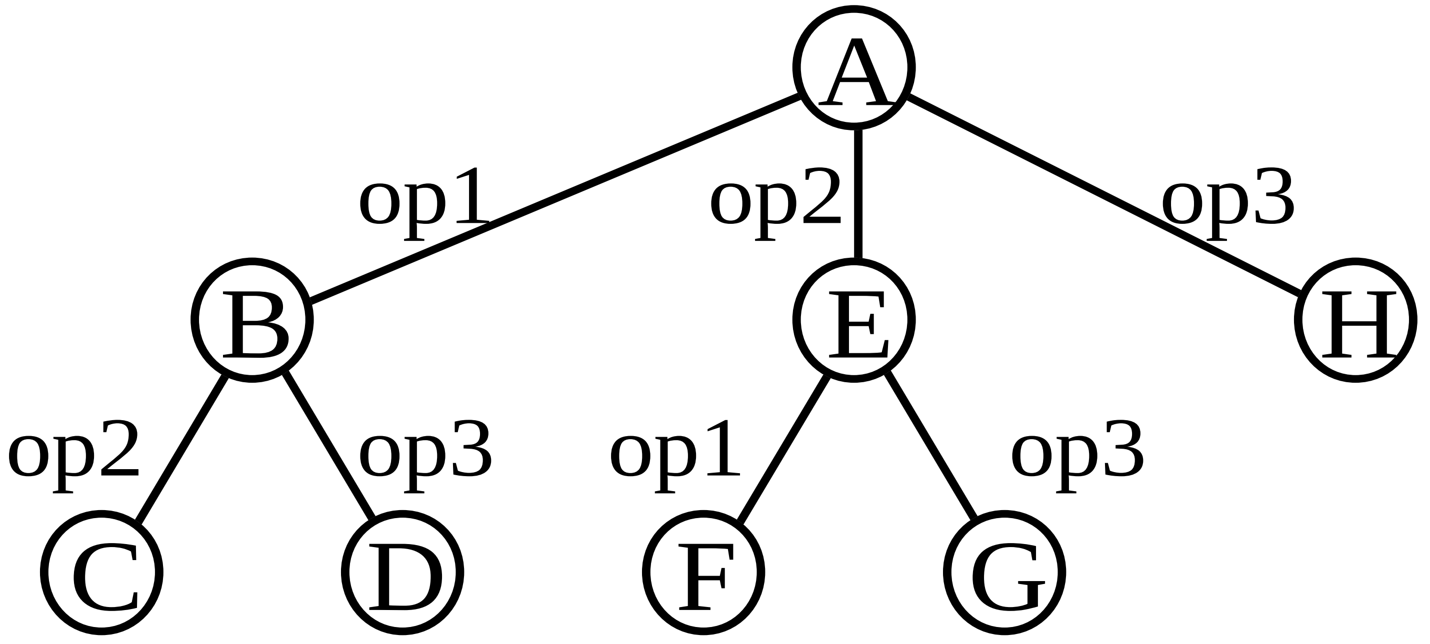
Rešenje

- Cilj pretrage je zadovoljen kada, primenom operatora, nije moguće generisati nova stanja
- Nije moguće više od dva puta uzastopno primeniti operatore
- => stablo ima dva nivoa

- A – startno stanje



- A, B, C, D, E, F, G, H

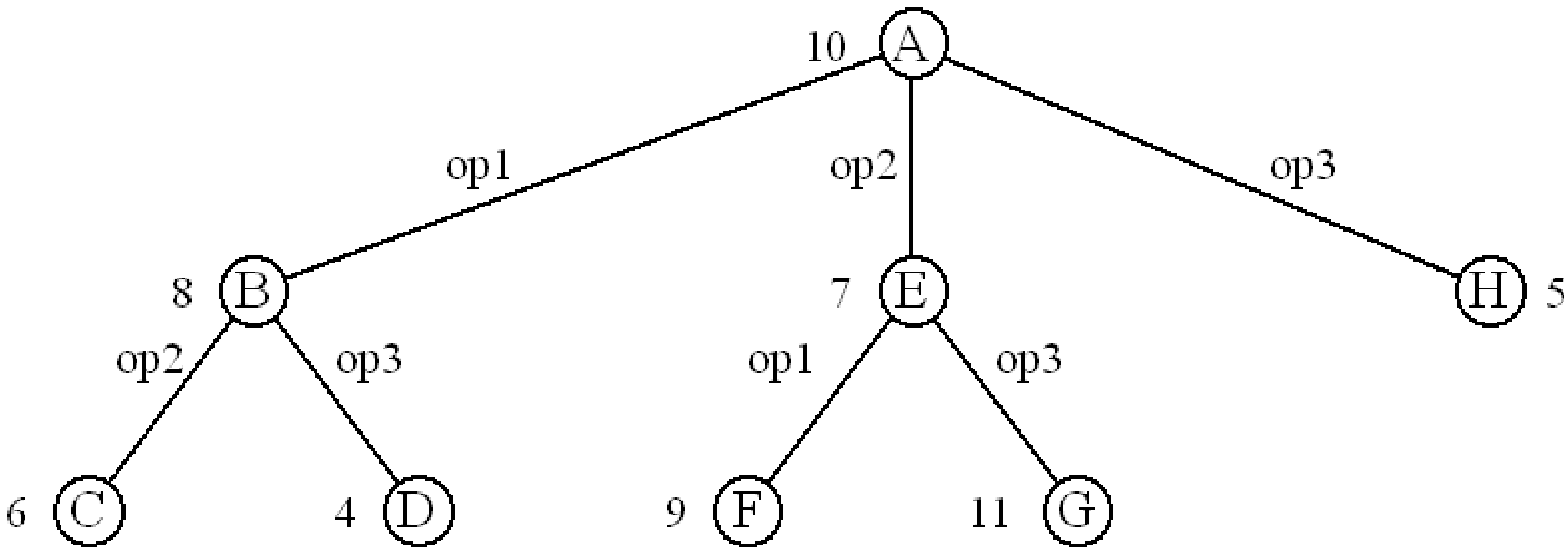


b) Ako se dati graf pretražuje metodom 'prvo najbolji' (*best-first*), navesti redosled obilaženja čvorova. Date su heurističke funkcije za svaki čvor, kao i sekvenca primene operatora koja vodi do čvora:

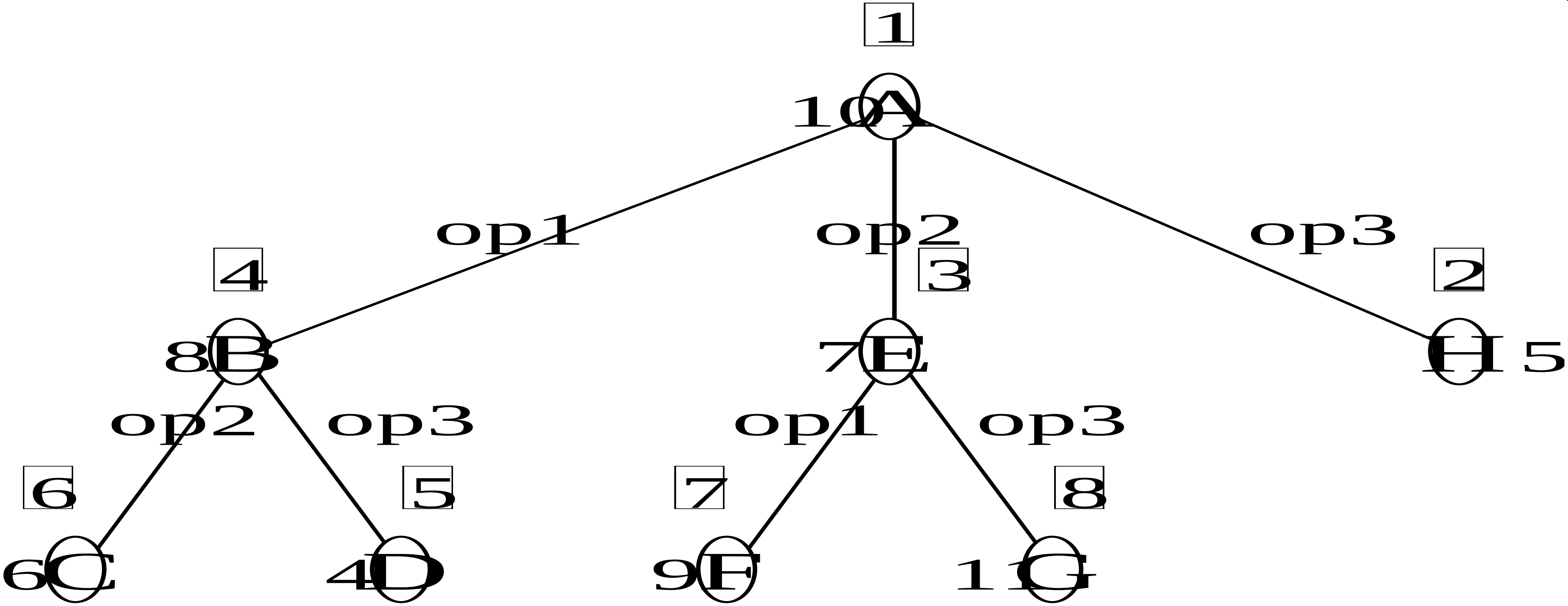
- C 6 - op1, op2
- D 4 - op1, op3
- F 9 - op2, op1
- G 11 - op2, op3
- B 8 - op1
- E 7 - op2
- H 5 - op3

A 10 - startno stanje

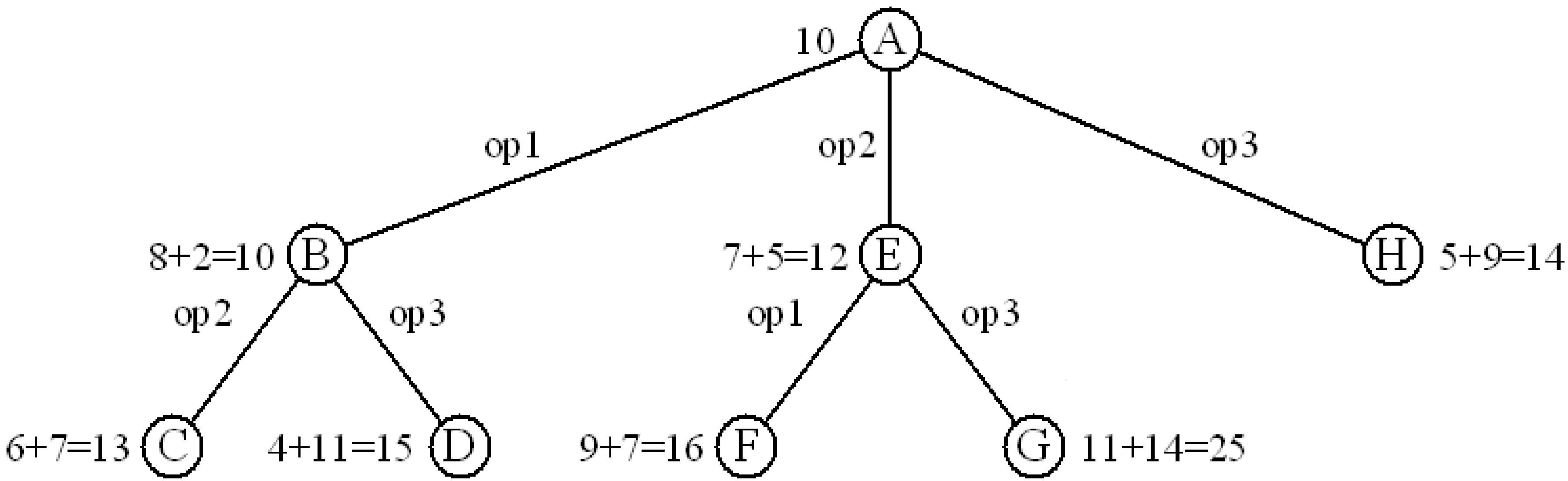
- Heuristička funkcija pored čvora

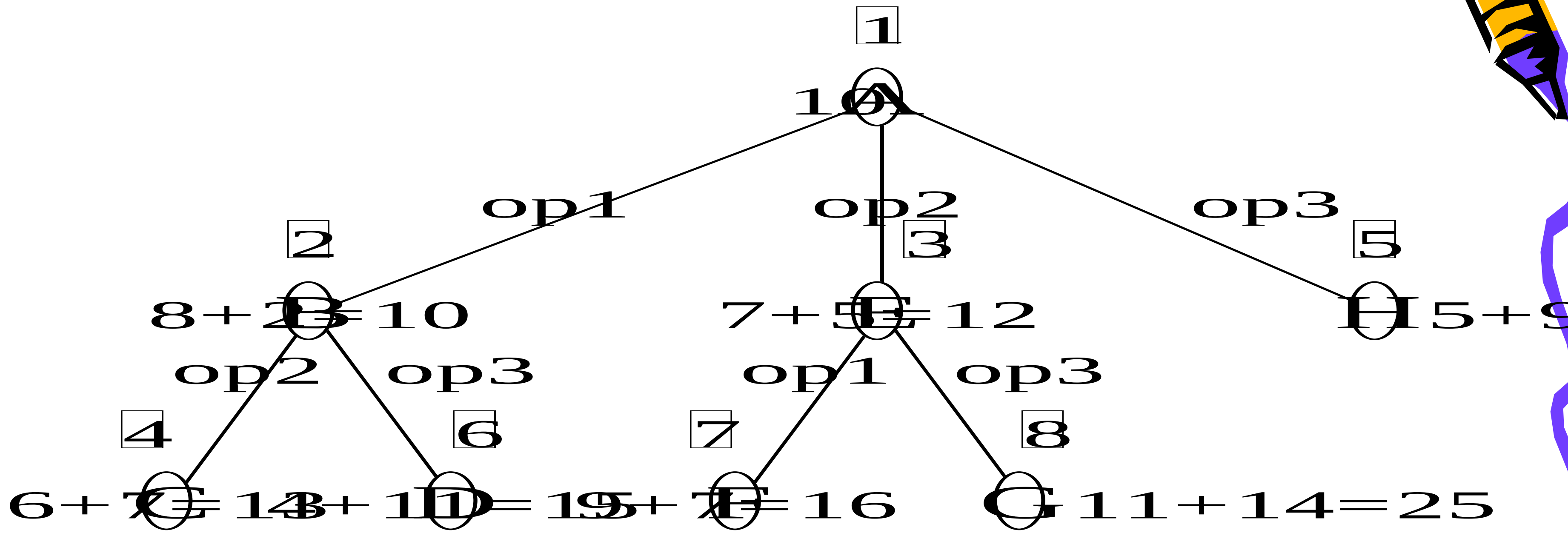


- A, H, E, B, D, C, F, G



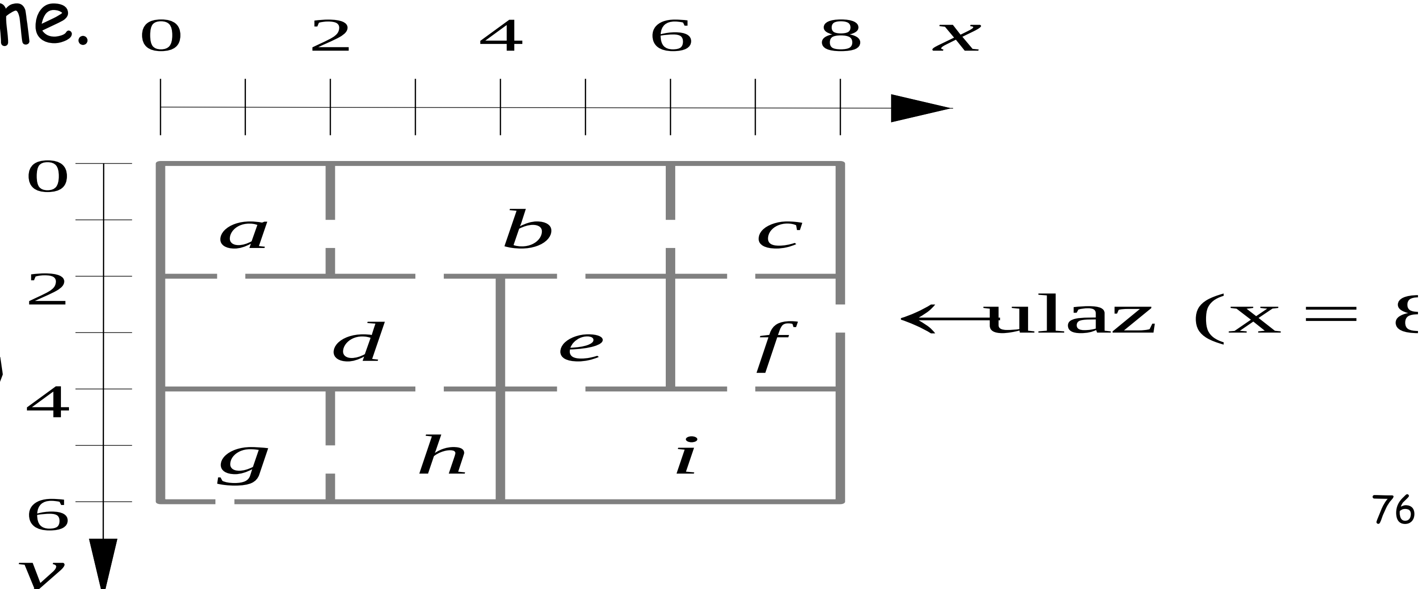
c) Ako bi se pretraživanje sprovelo metodom *A**, polazeći od heurističkih funkcija datih pod b), navesti redosled pretraživanja ako su cene primene operatora : 2 za *op1*, 5 za *op2*, i 9 za *op3*.


10
A
op1
op2
op3
8+2=10
B
7+5=12
E
H
5+9=14
op2
op3
op1
op3
6+7=13
C
4+11=15
D
9+7=16
F
G
11+14=25

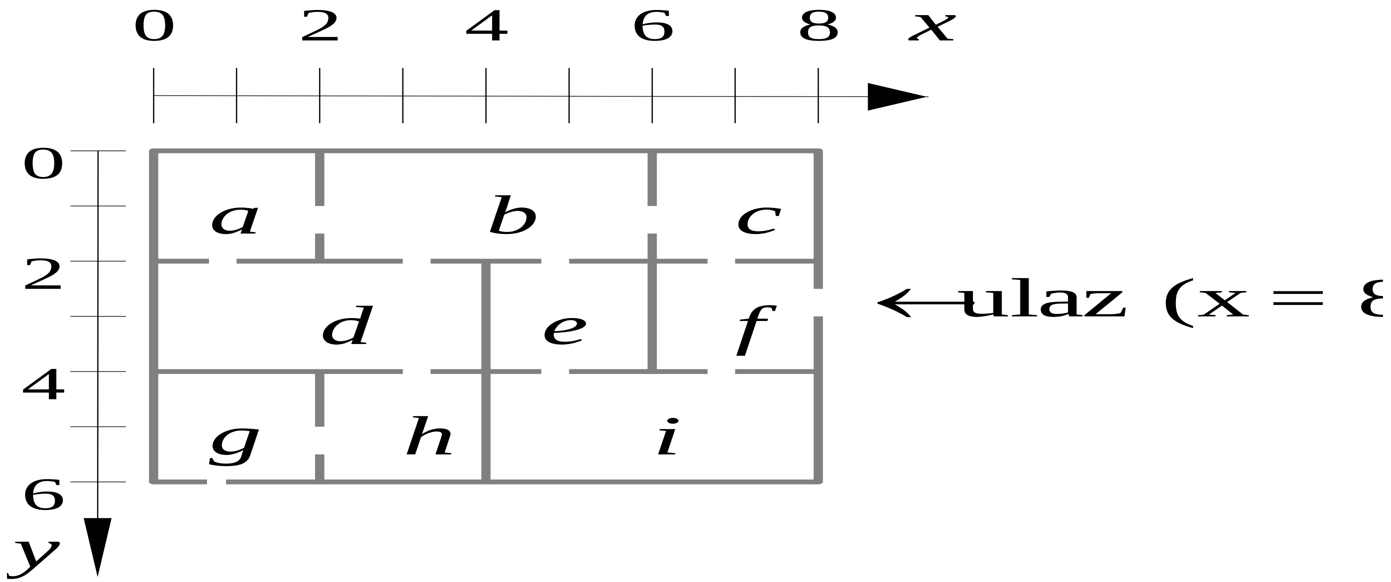

1
10
A
op1
op2
op3
2
3
5
8+2=10
B
7+5=12
E
H
op2
op3
op1
op3
4
6
7
8
C
D
F
=16
G
11+14=25

Zadatak 6: Džems Bond

Na slici je prikazan tlocrt jedne kuće, u kojoj se, u prostoriji *g*, nalazi garaža u kojoj je auto Aston Martin (koordinate: x=1, y=5). U kuću utrčava Džems Bond, 8 sekundi ispred grupe loših momaka koji ga jure, a svi se kreću brzinom od 1 *m/s*. Dž.B. poseduje senzor koji mu javlja udaljenost od auta a loši momci pretražuju kuću deleći se u manje grupe po potrebi, tako da svaku prostoriju zaposedaju za najkraće moguće vreme.



a) Koji metod pretrage koristi Dž.B., a koji loši momci?



- Tlocrt predstavlja graf pretrage
- Dž. B. kao pojedinac vrši izbor lokalno najboljeg naslednika
- Senzor pomaže za odredjivanje rastojanja
- => ?

- Metod planinarenja

- Loši momci ne poseduju senzor
- => nema heurističke funkcije!
- Optimalna putanja
- => ?

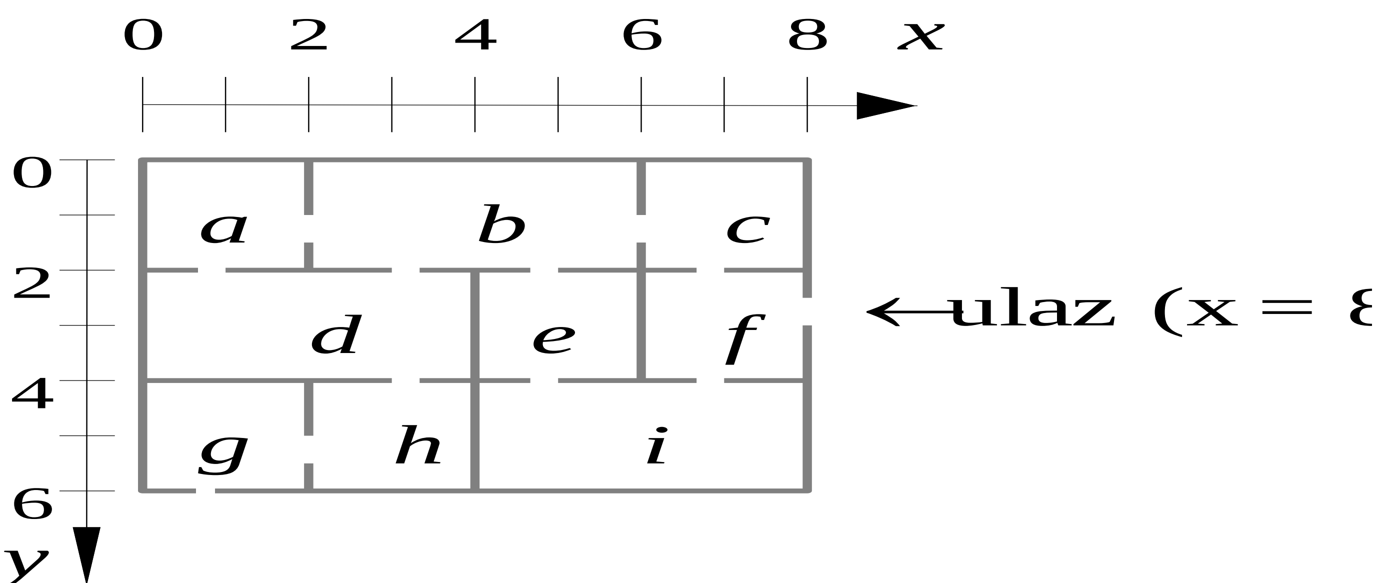
- Metod grananja i ograničavanja

\+

- Dodatna pretpostavka: neko ostaje u obidjenim prostorijama

  => dinamičko programiranje

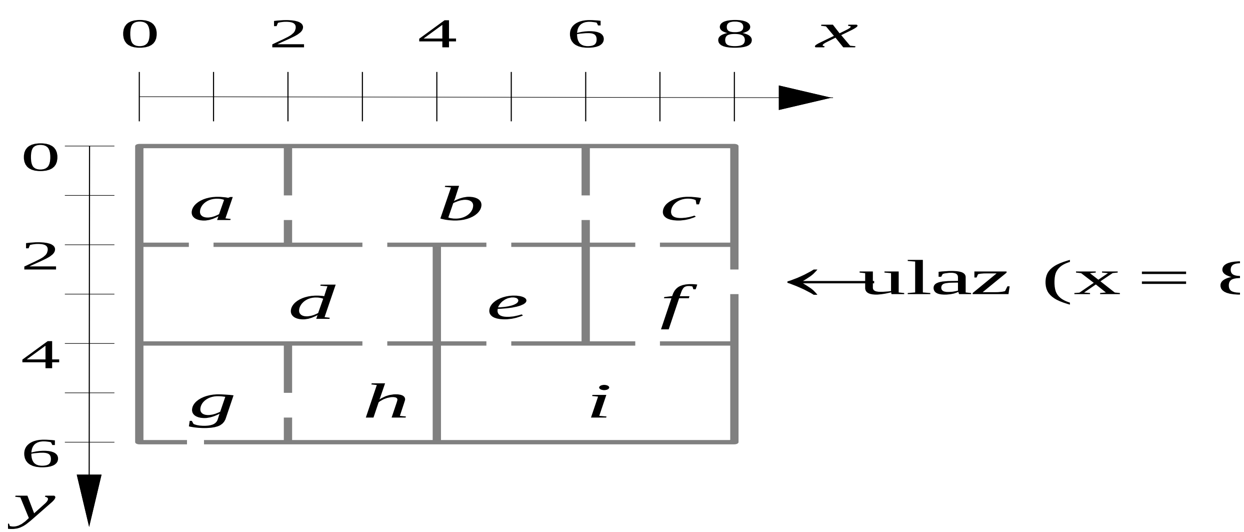
b) Pokazati kako će se kretati Dž.B., a kako njegovi gonioci, i odrediti hoće li Dž.B. uspeti da stigne do auta pre loših momaka.



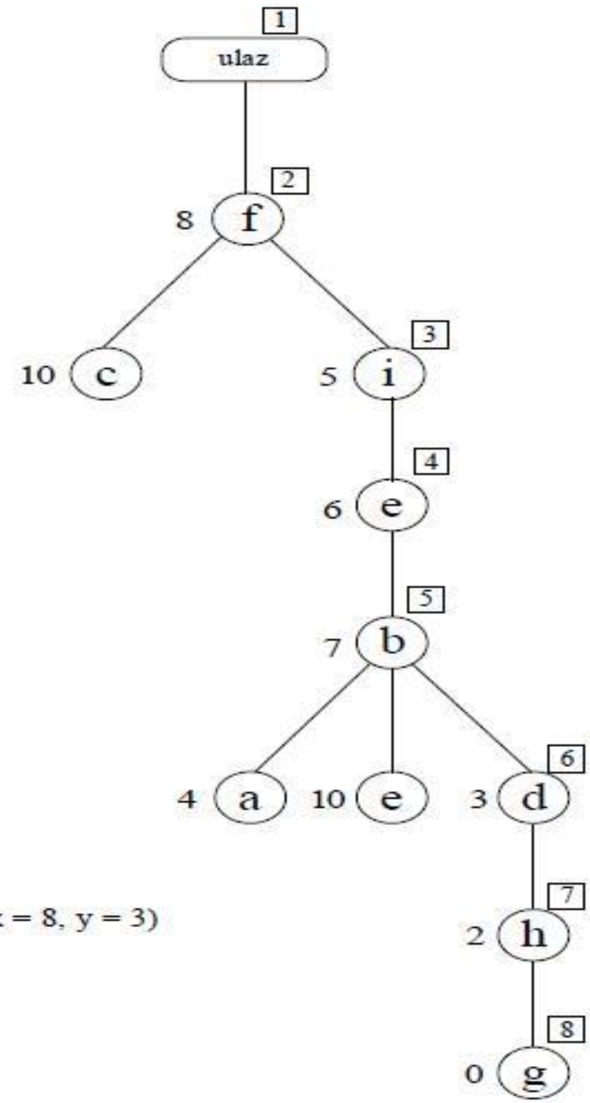
- Uspešno će umaći ukoliko stigne do kola a ne sretne ni u jednoj od soba gonioce
- Potrebno je odrediti redosled soba i vremenske trenutke boravka u njima
- Uporediti sa vremenskim trenucima kada sobe bivaju okupirane od strane gonilaca

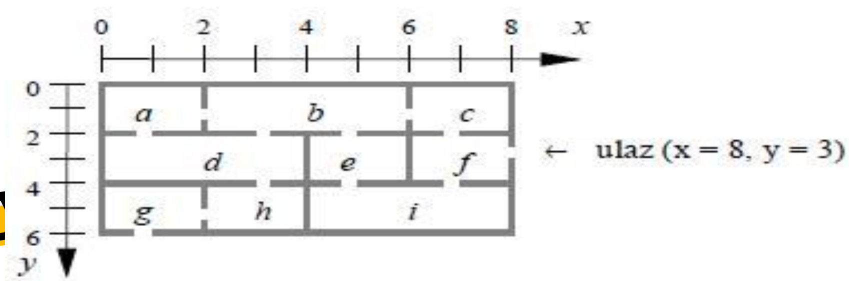
Pretpostavke

- Kretanje je isključivo u pravcu x ili y ose (nema dijagonalnog kretanja)
- Pri prelaženju iz sobe u sobu uvek se ide iz centra jedne u centar druge sobe



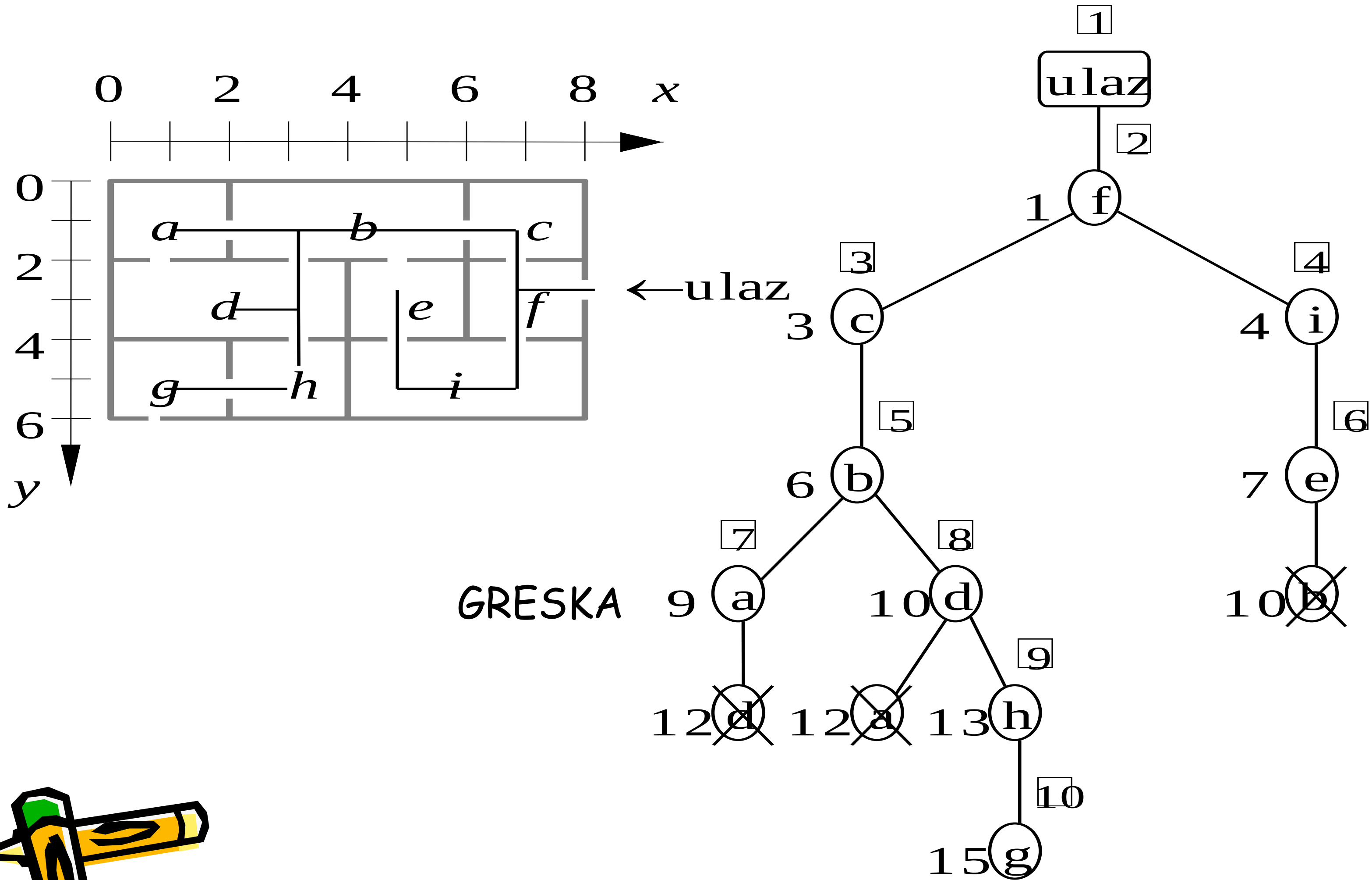
- Kretanje Dž. B.





- f = 0s - 2s
- i = 2s - 6s
- e = 6s - 8s
- b = 8s - 12s
- d = 12s - 16s (ide se do centra sobe)
- h = 16s - 18s
- g = 18s - 19s, kada ulazi u kola i velikom brzinom izvozi ih iz kuće

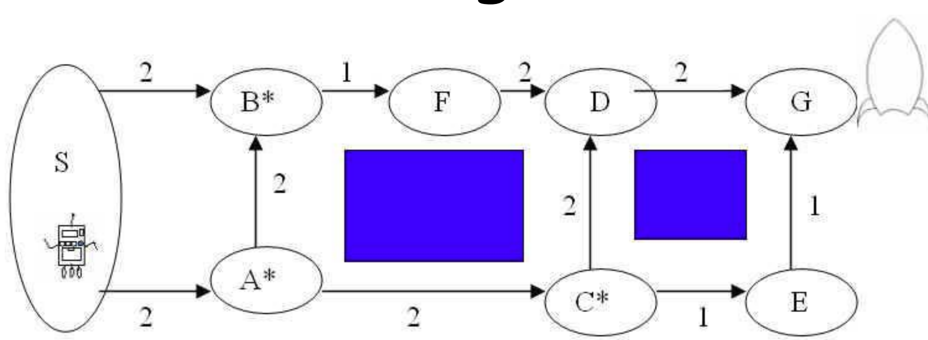
- Kretanje loših momaka

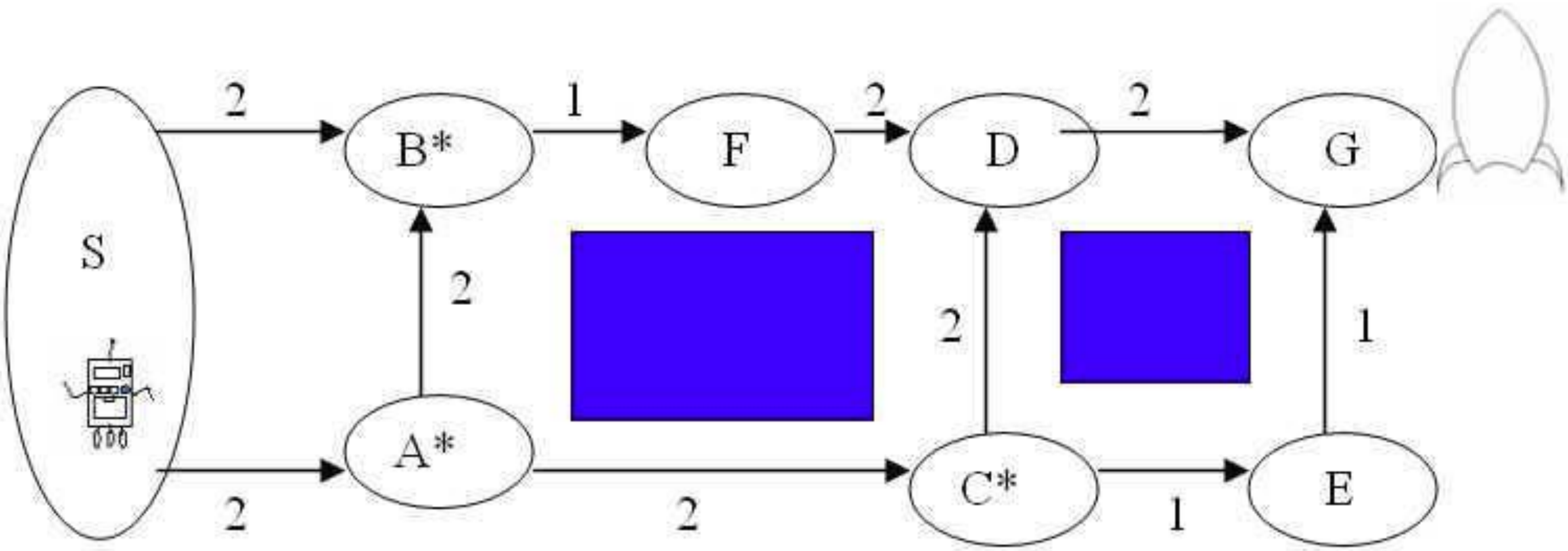


Aston Martin

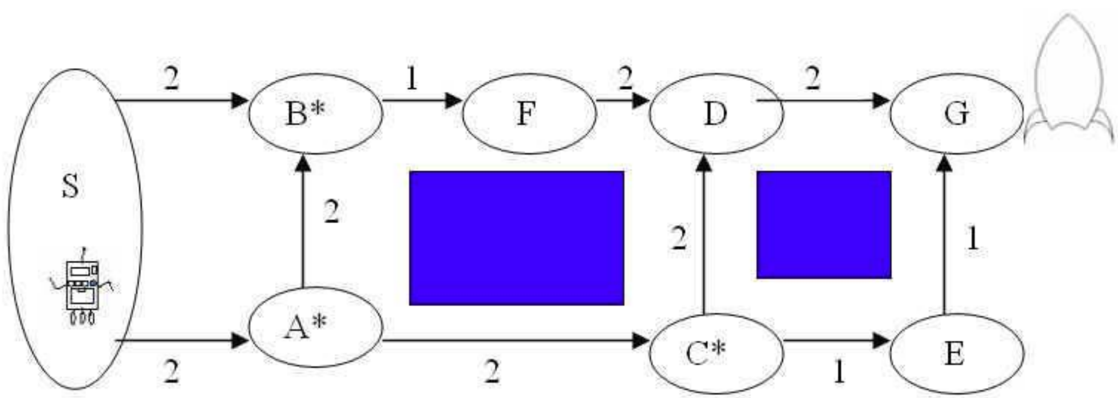
Zadatak 7: Povratak sa meseca

Prof. Boško i Gliga su upravo završili obilazak Meseca i planiraju da se vrate nazad na Zemlju u njihovom vasionskom brodu (smešten na lokaciji G). Lokalni robot ima veliku želju da krene sa njima, ali mora da požuri kako bi na vreme stigao do vasionskog broda iz tačke S, gde se trenutno nalazi

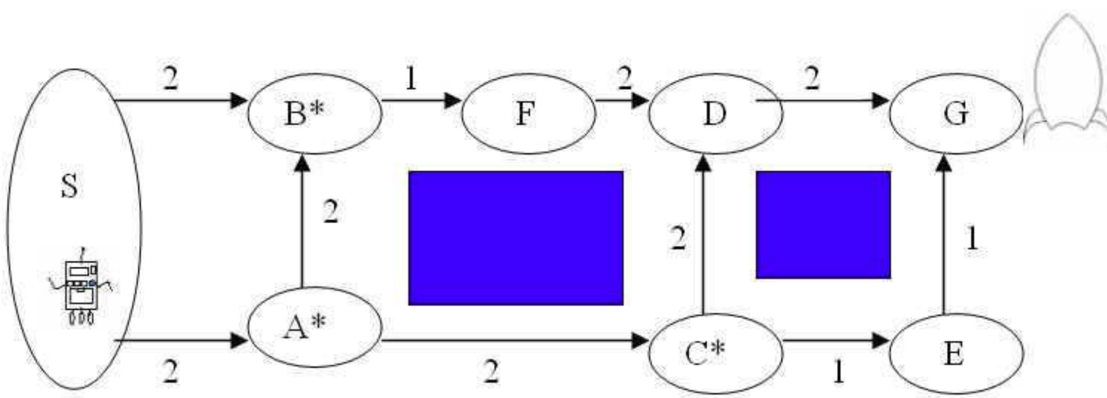



S
B*
F
D
G
A*
C*
E
2
1
2
2
2
2
1
2
2
1

- Prilikom kretanja mora da obrati pažnju na navigaciju. Navigacija mora da bude pametna da bi izbegao semafore, jer ga oni usporavaju bilo da ide pravo ili skreće ulevo (skretanje u desno ga ne usporava).



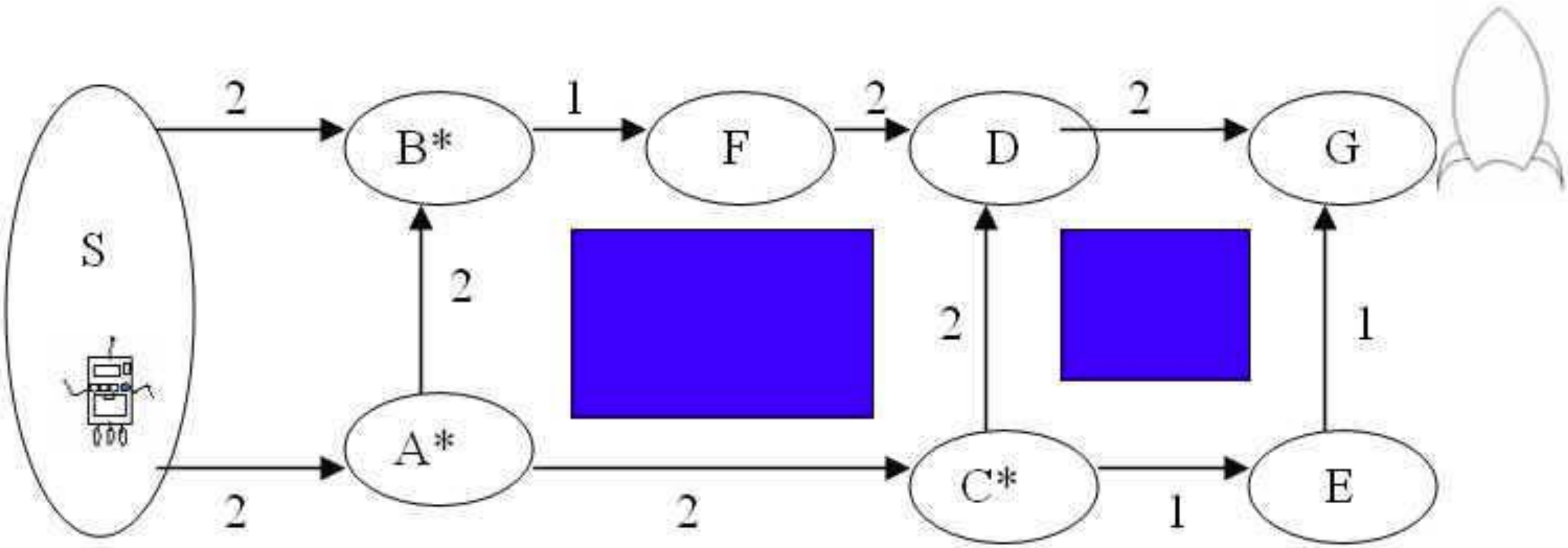
- Postoje tri semafora oko zgrada (zgrade su zadate pravougaonicima). Semafori se nalaze na raskrsnicama A, B, C (što je obeleženo sa *).
- Dužine puteva su obeležene na slici kao i mogući pravci kretanja



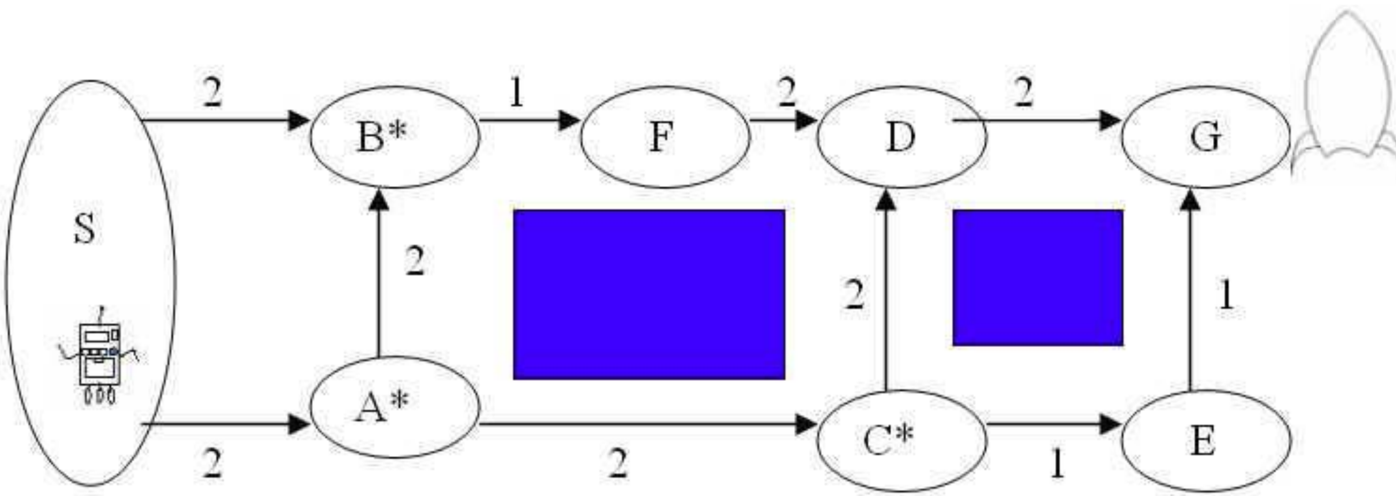
- Imamo sledeća pravila za semafore
  - Ukoliko se ide pravo semafor oduzima vreme ekvivalentno produžetku putanje za 1 (nema sreće)
  - Ukoliko skreće 90 ulevo semafor oduzima vreme ekvivalntno produžetku putanje za 2 (očigledno, najpre mora da sačeka zeleno a zatim da se gužva rasčisti)
  - Skretanje udesno nema uticaja na vreme kretanja

- Vrednosti heurističke funkcije od pojedinih čvorova do čvora G su:

S=5, A=3, B=2, C=3, D=1, F=2, E=3, G=0



a) Nacrtati stablo pretrage, uredjujući naslednike svakog čvora alfabetski sa leva na desno



b) Pretraga po dubini

c) Pretraga po širini sa pamćenjem obidjenih čvorova

d) Metod planinarenja

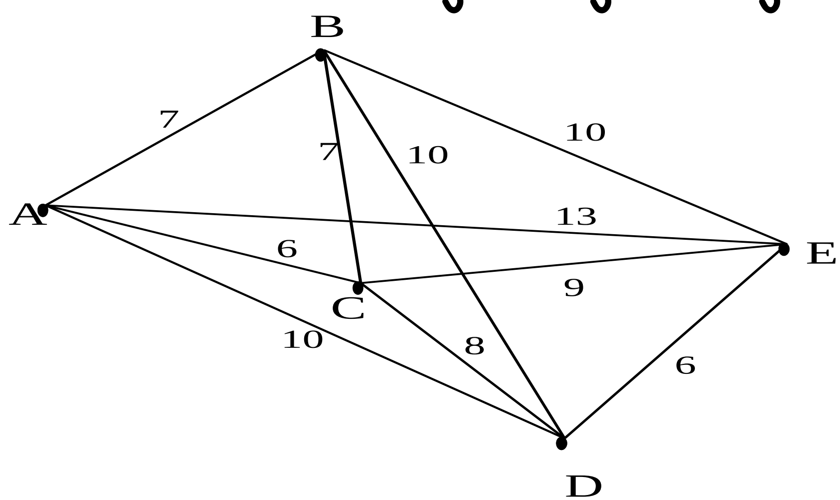
e) Metod grananja i ograničavanja

f) A*

...

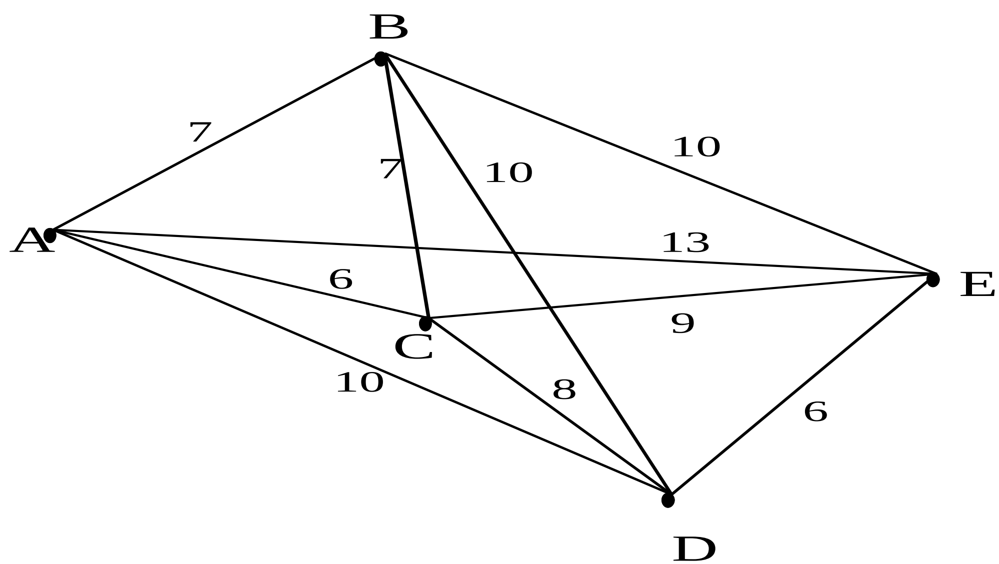
Zadatak 8: Problem trgovačkog putnika

Trgovački putnik mora da poseti svaki od pet gradova prikazanih na slici. Između svakog para gradova postoji put, dužine naznačene na slici. Polazaći od grada A, naći minimalan put koji obezbeđuje posetu svakom gradu samo jedanput i povratak u A. Predložiti dve različite heurističke funkcije. Za svaku od funkcija primenom nekog od algoritama pretraživanja naći rešenje problema. Koja od predloženih funkcija daje bolje rešenje?

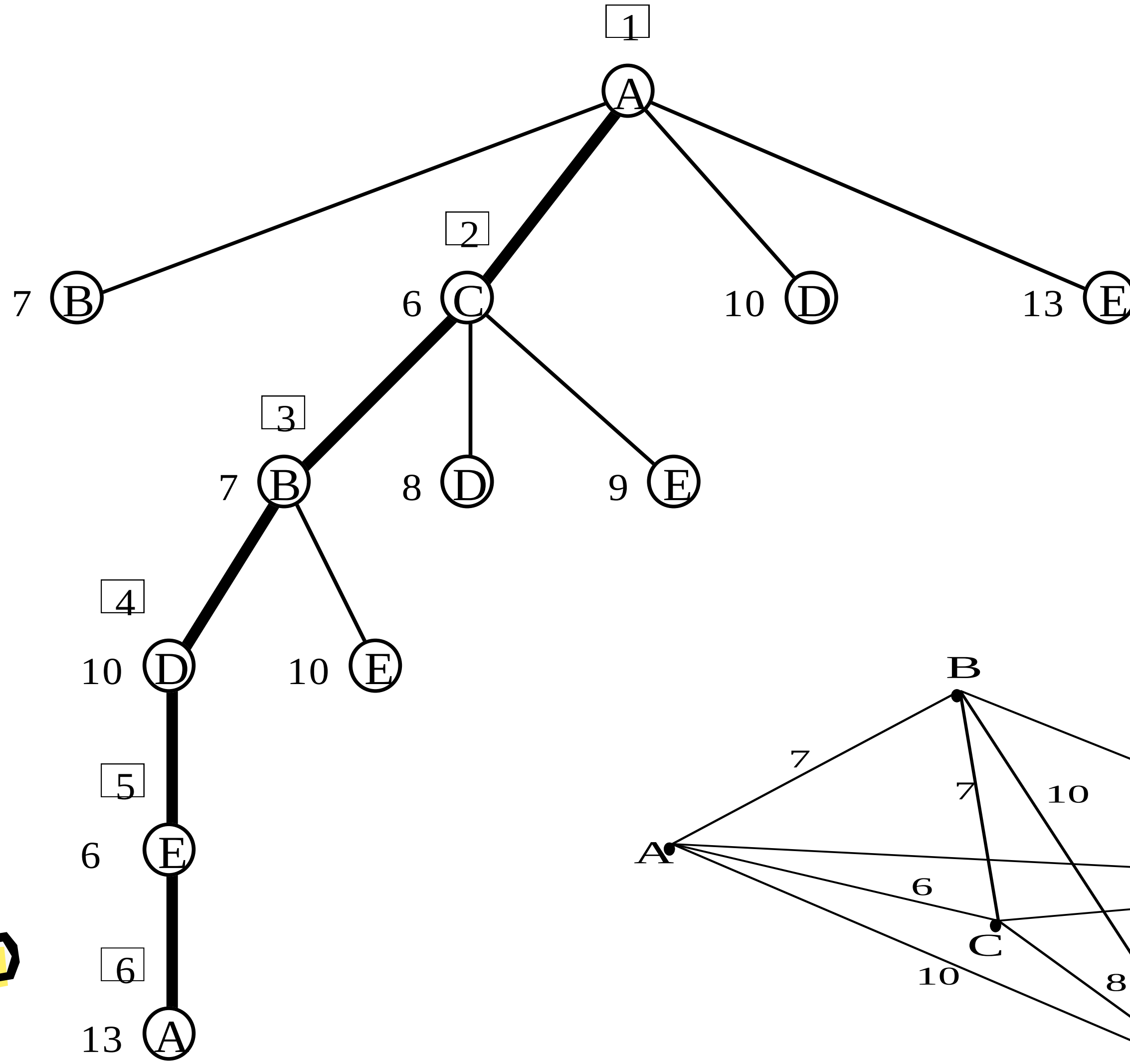


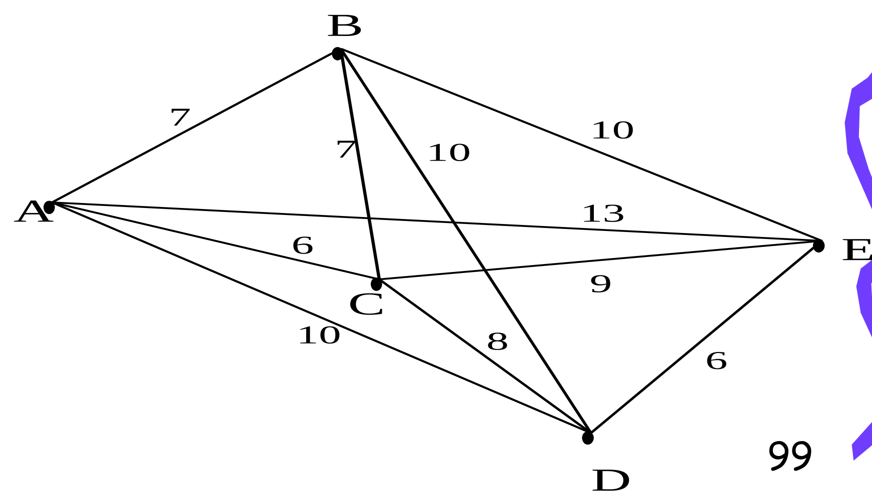
Heuristika h1

- Prioritet se daje neobidjenom gradu koji je najbliži tekućem gradu
- h1 = rastojanje
- Ocenjuje čvor lokalno



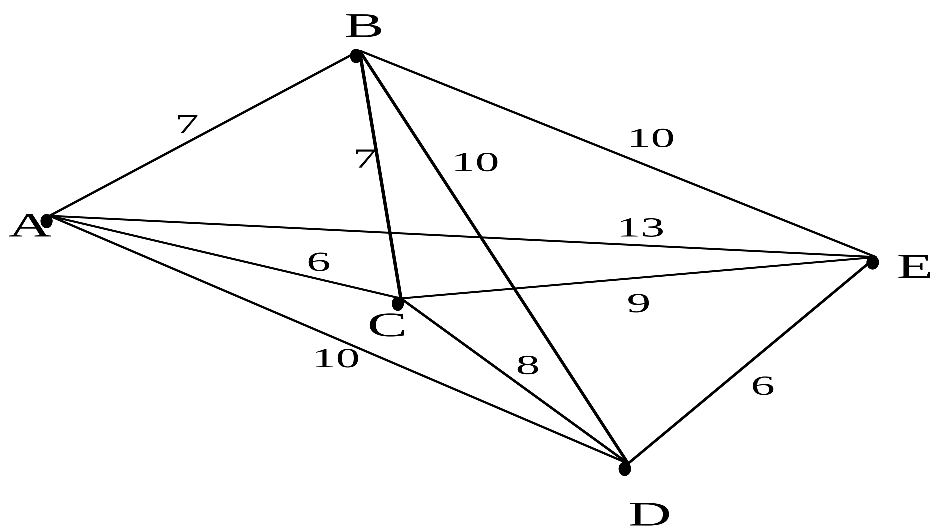
- Metod planinarenja





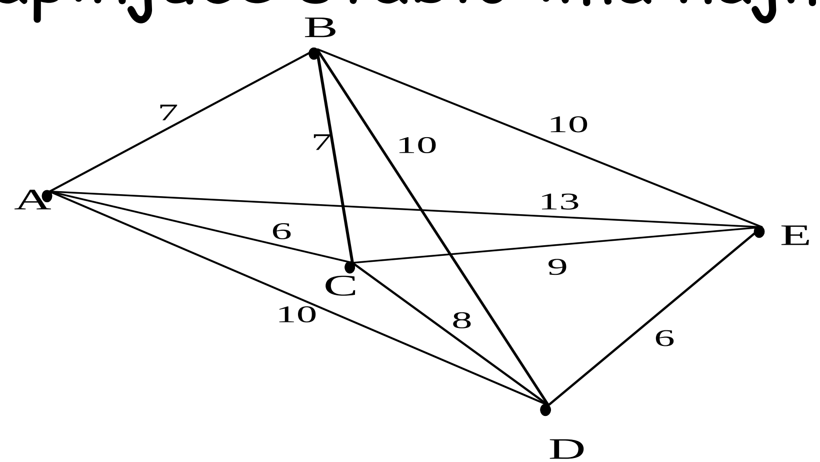
Heuristika h2

- h2 = dužina minimalnog razapinjućeg stabla, koje obuhvata sve čvorove grafa koji se ne nalaze na putanji (startni i ciljni čvor ne treba uklanjati)



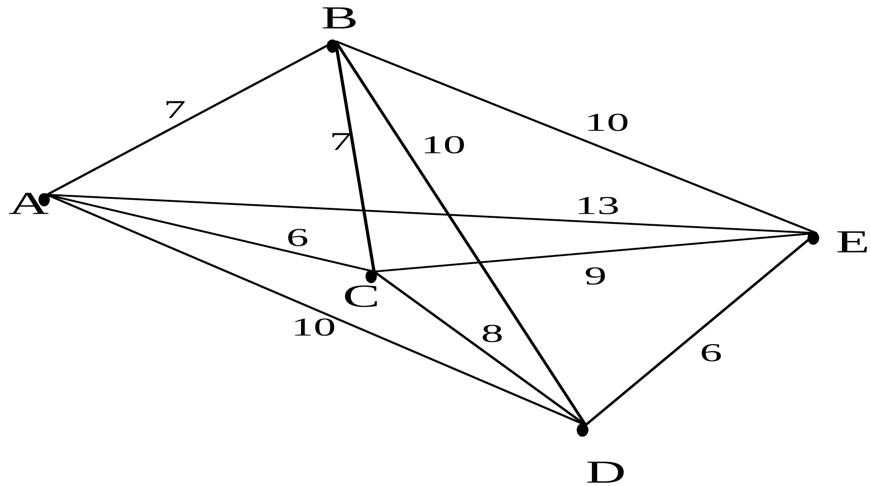
Minimalno Razapinjuće Stablo

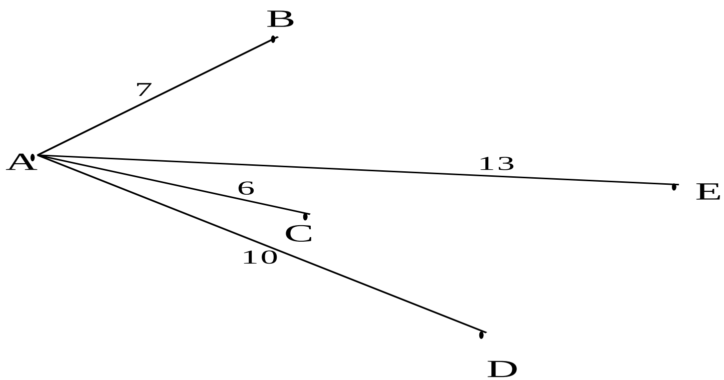
- Razapinjuće stablo u nekom grafu je skup grana grafa koje povezuju čvorove grafa tako da nije formirana nijedna zatvorena putanja
- Dužina razapinjućeg stabla je zbir dužina svih grana koje ga sačinjavaju
- Minimalno razapinjuće stablo ima najmanju dužinu



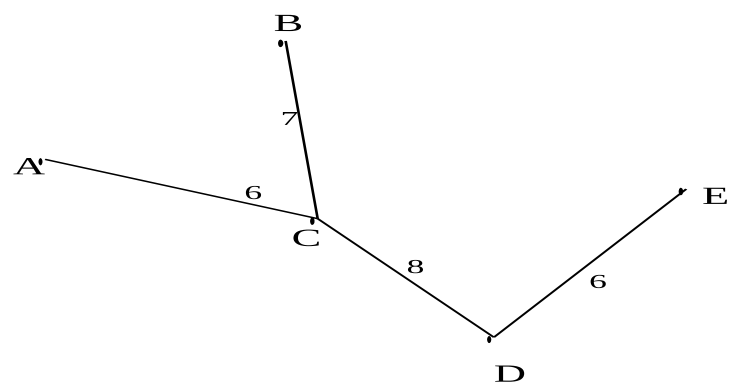
Konstrukcija minimalnog razapinjućeg stabla

- Na počektu u stablu se nalaze svi čvorovi i nijedna grana
- U stablo uključujemo najkraću granu koja već nije uključena i čijim se uključivanjem ne kreira petlja u stablu
- Prethodni korak se ponavlja dok se ne povežu svi čvorovi u jedinstveno stablo

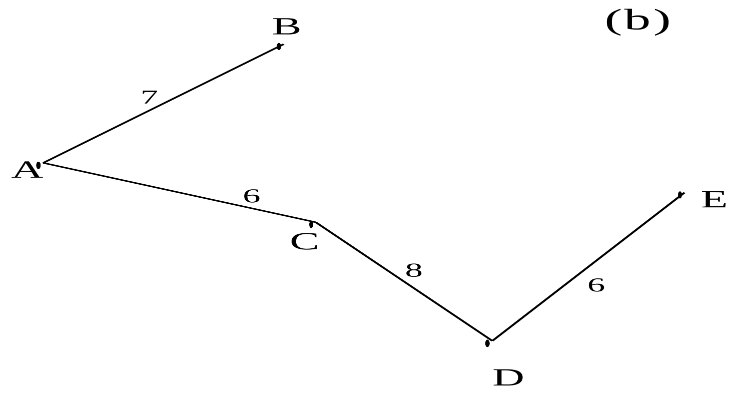




(a)

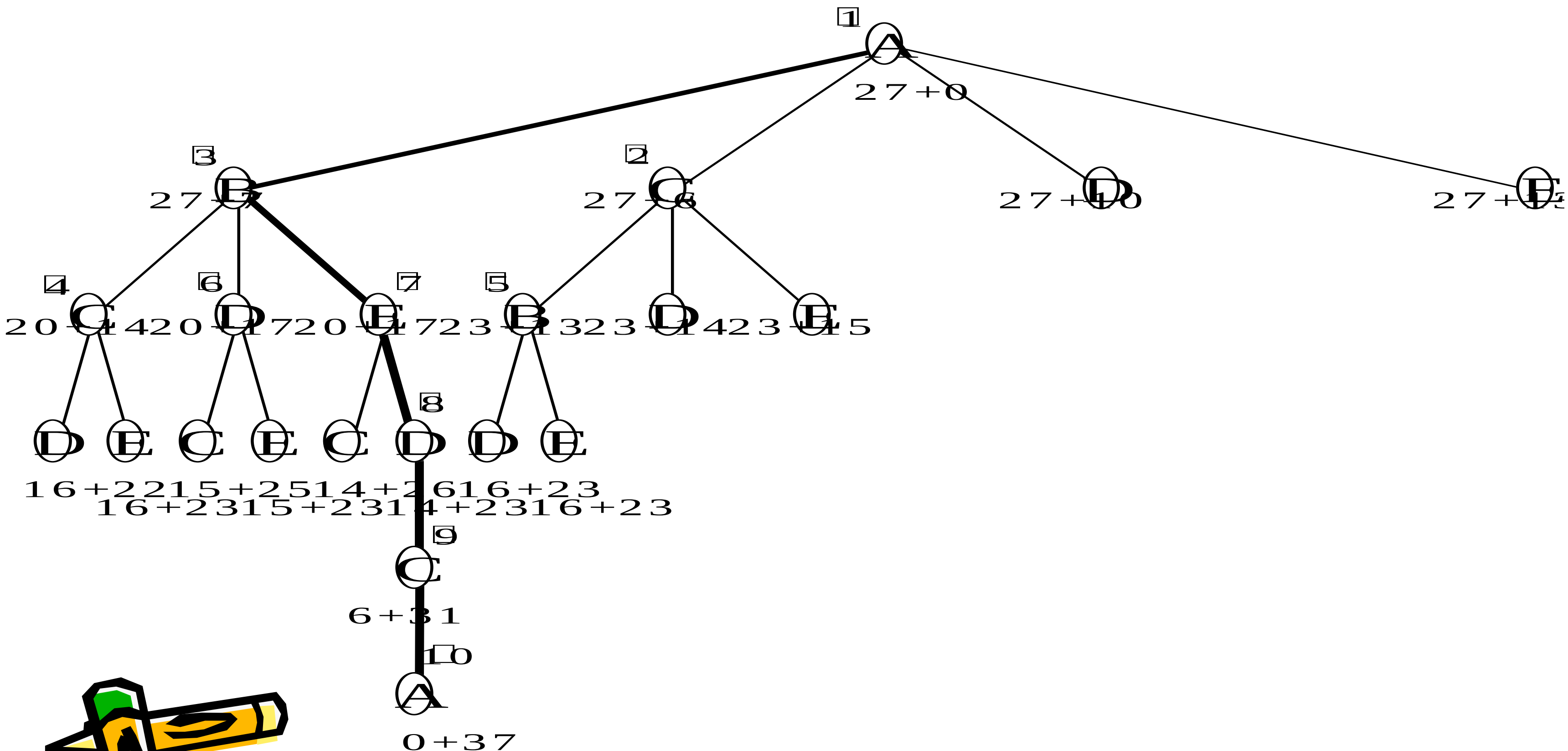


(b)



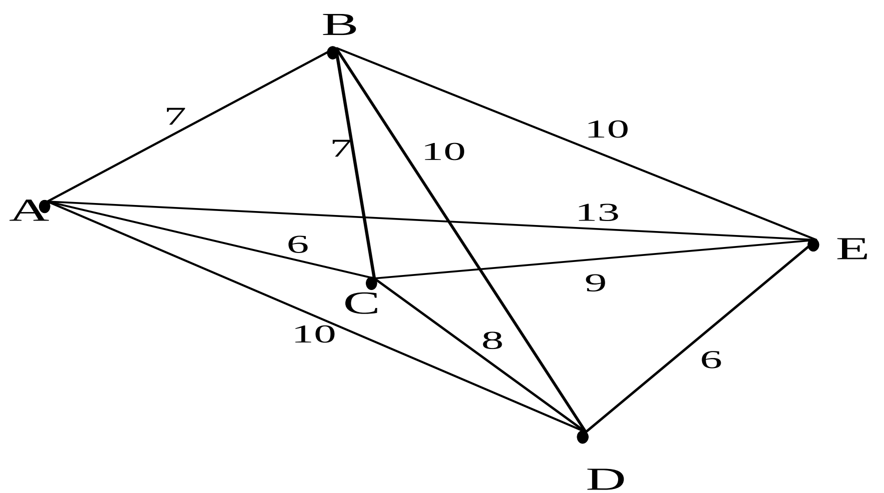
(c)

- Heuristička funkcija + cena koštanja



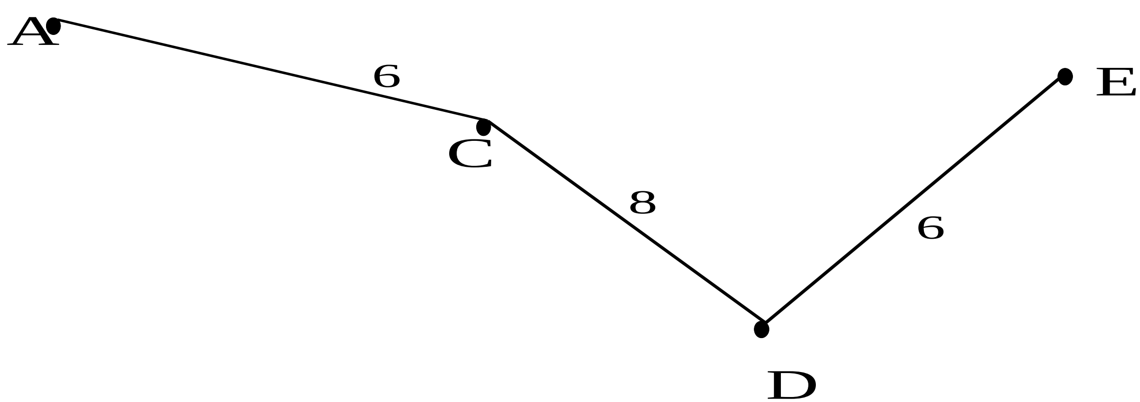
Primer odredjivanja funkcije procene

- Putanja A, B, E
- h2(E)= dužina MRS({A,C,D,E})



Primer odredjivanja funkcije procene

- Dužina parcijalne putanje c(A-B-E) = rastojanje(A,B) + rastojanje(B,E) = 7 + 10 = 17 tako da je ukupna vrednost funkcije f za čvor E jednaka 37.



Zadatak 9: Problem zamenjivanja brojeva

Data su sledeća pravila koja se mogu iskoristiti za zamenu brojeva na levoj strani nizom brojeva na desnoj strani:

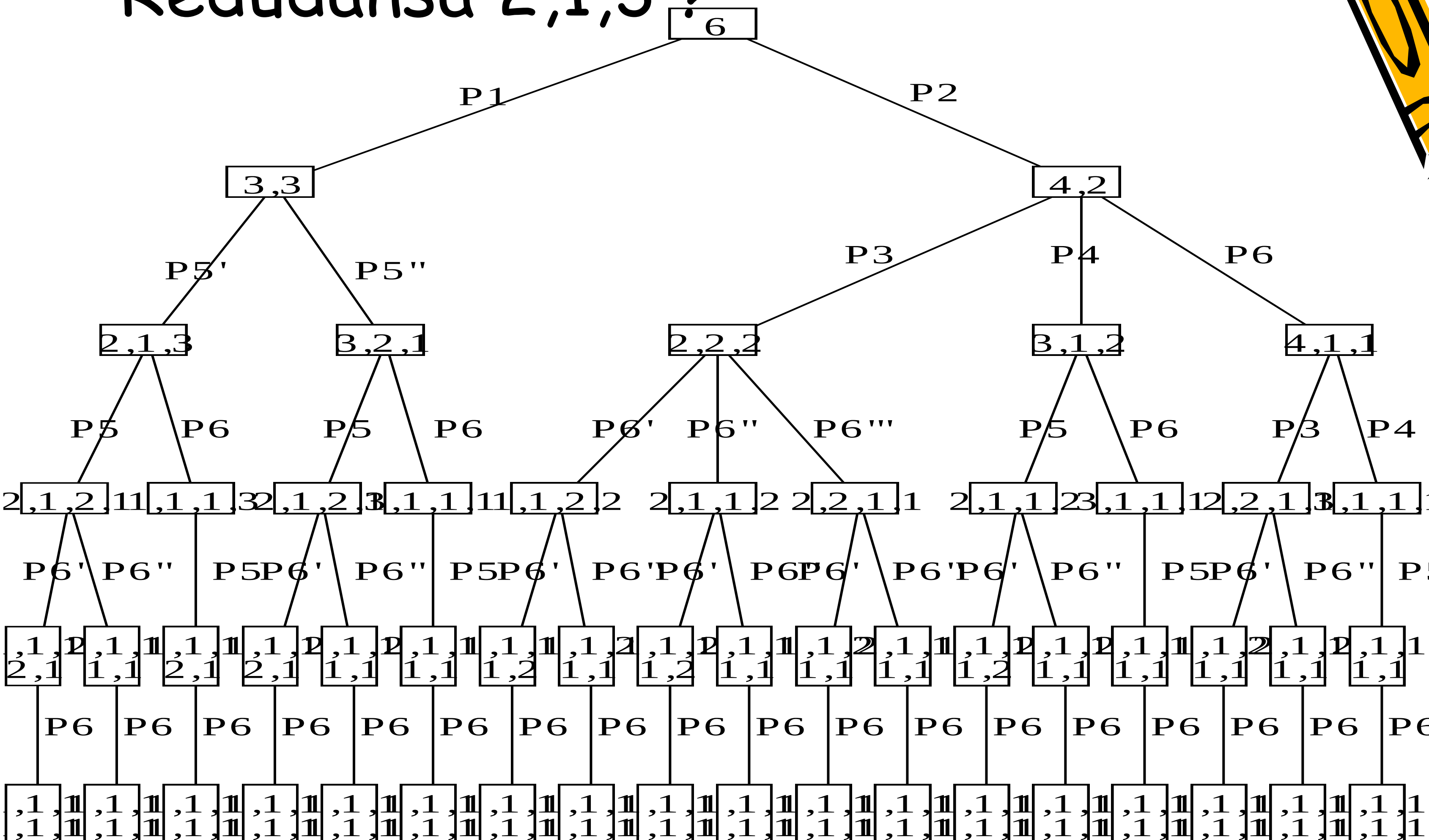
P1: 6 -> 3,3 P3: 4 -> 2,2 P5: 3 -> 2,1

P2: 6 -> 4,2 P4: 4 -> 3,1 P6: 2 -> 1,1

Kako se mogu iskoristiti ova pravila da se broj 6 transformiše u niz jedinica? Pokazati kako AO* algoritam obavlja ovu transformaciju. Usvojiti da je cena k-konektora k jedinica, a vrednost heurističke funkcije h u čvoru označenom brojem 1 je nula a čvora označenog sa n iznosi n

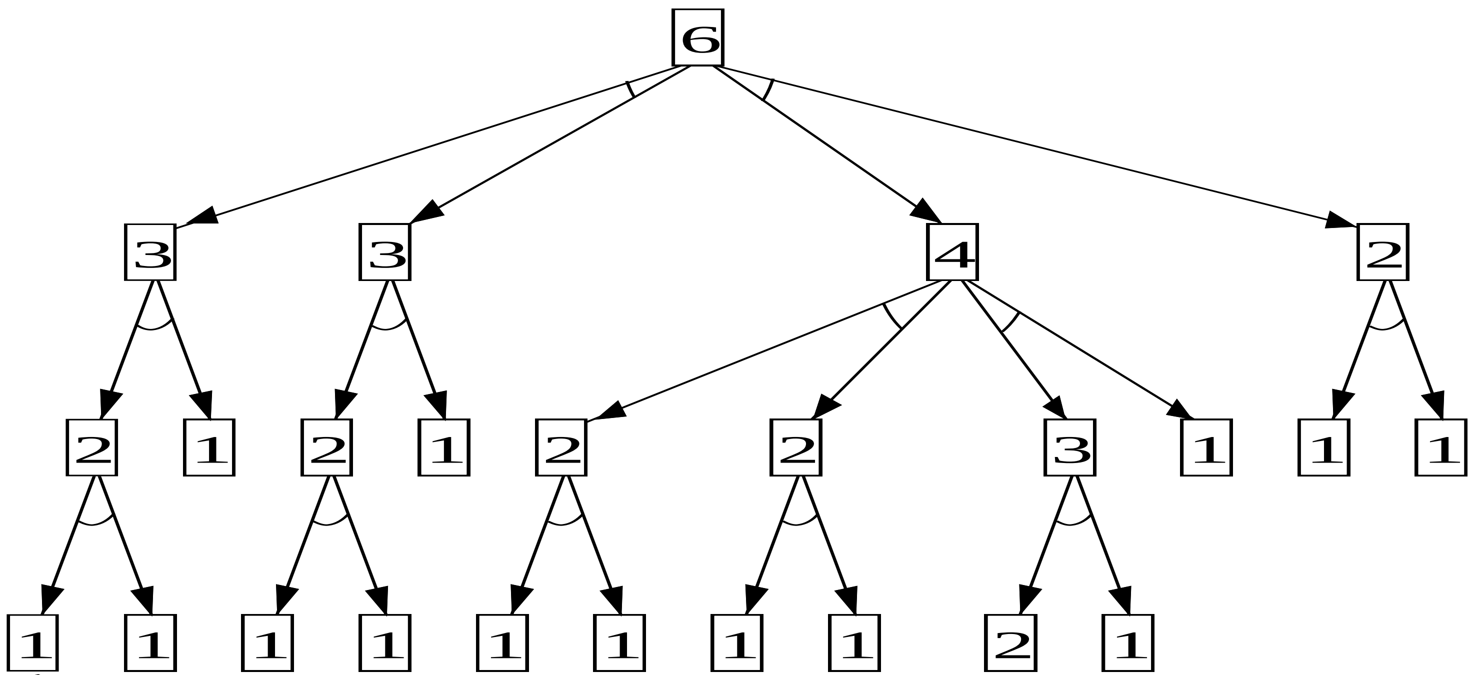
- Klasična pretraga ?

- Redudansa 2,1,3 ?

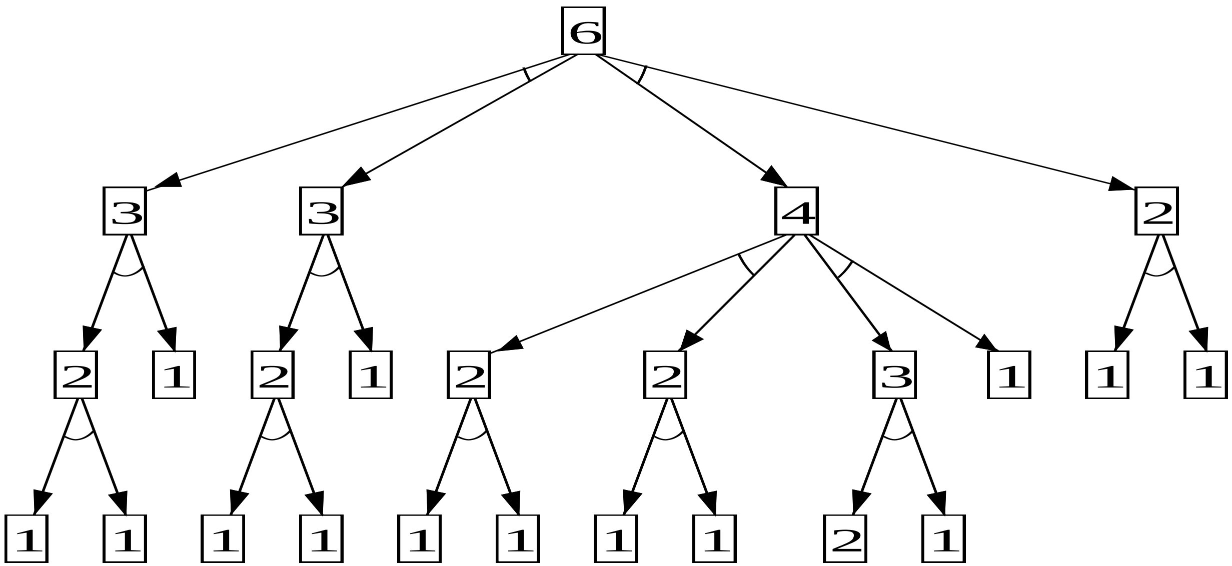


- Zadati problem rastavimo na niz nezavisnih potproblema
- Zamena jedne cifre jedinicama je nezavisna od drugih cifara u istom stanju
- Pr: Prelazak iz 4,2 u stanje 1,1,1,1,1,1 posebno posmatramo zamenu 4 jedinicama, a posebno zamenu 2 jedinicama

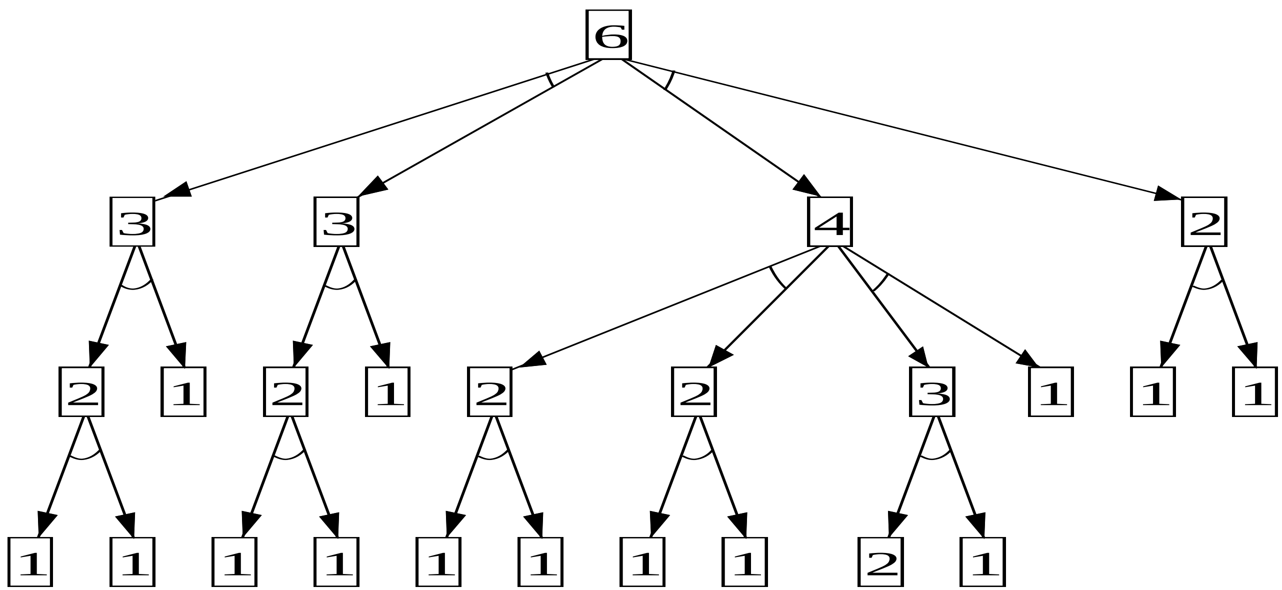
AND-OR stablo



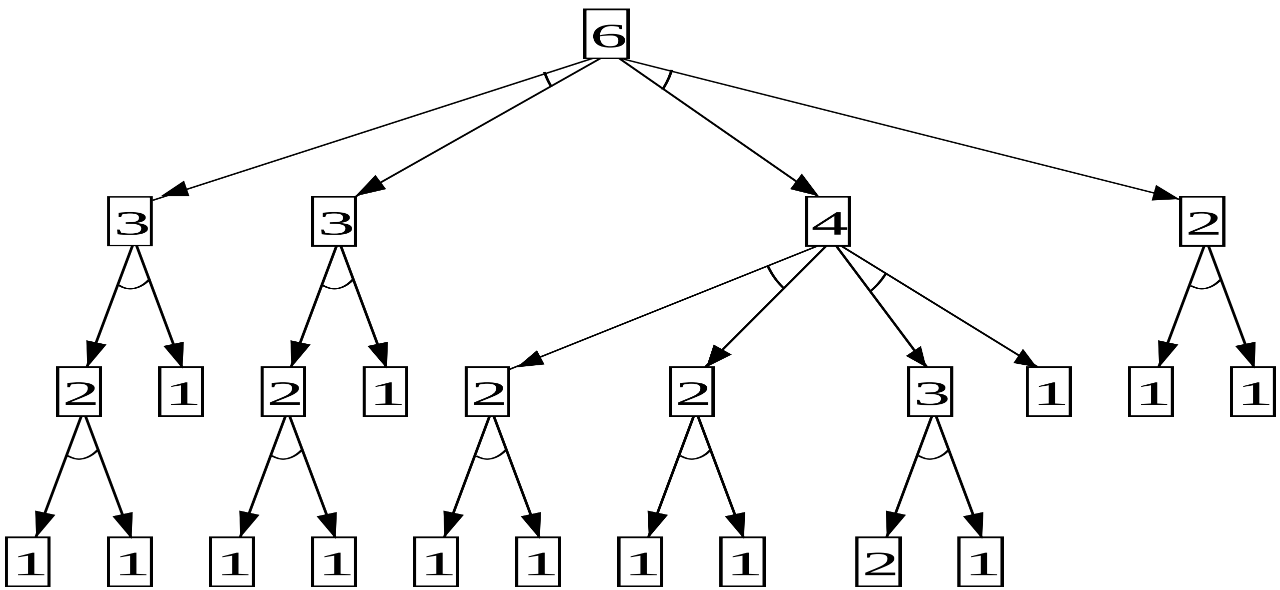
- Čvorovi stabla su potproblemi
- Čvorovi su povezani k-konektorima (grane koje povezuju jednog roditelja sa k čvorova naslednika)
- Grane koje čine konektor povezane su lukom



- Iz čvora 6 polaze dva 2-konektora
- Levi: povezuje čvor 6 sa dva čvora, oba označena sa 3 (pravilo P1)
- Desni: povezuje čvor 6 sa čvorovima 4 i 2 (pravilo P2)

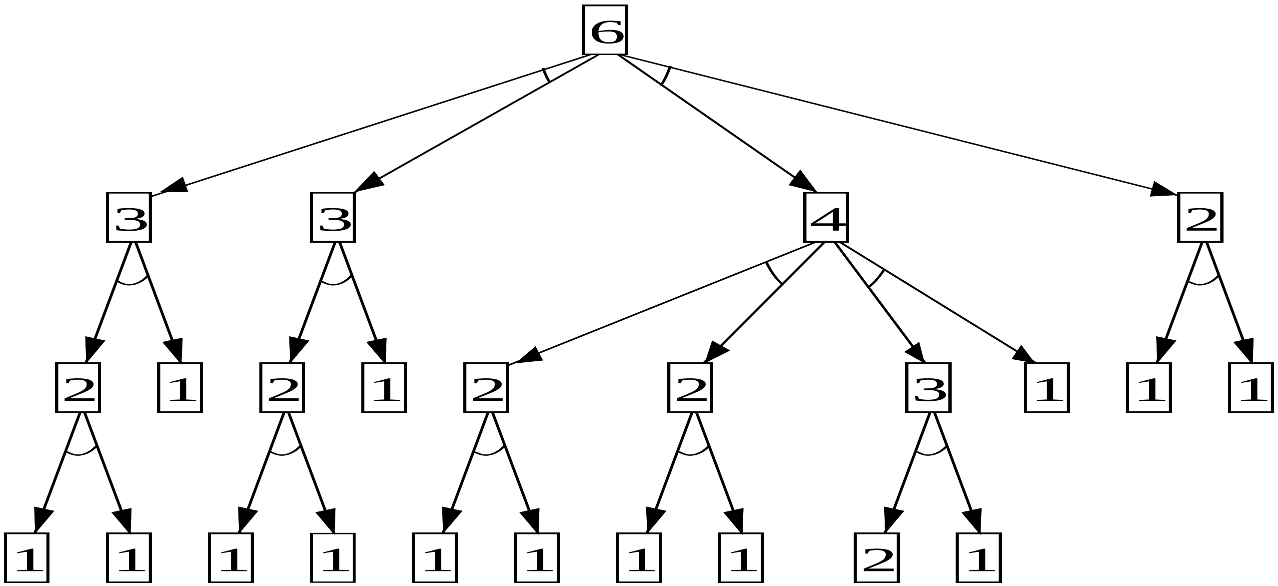


- Cifra 6 može se zameniti ILI sa
  - 3 I 3
  - 4 I 2
- Potproblem je rešen ako je rešen primenom bilo kog konektora

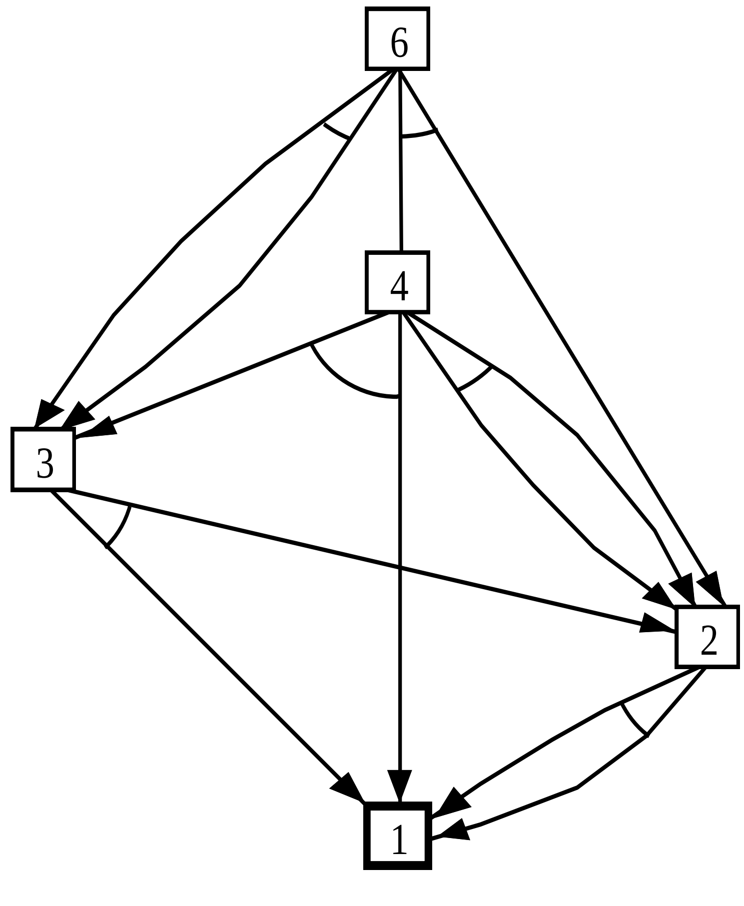


- Konektor => I relacija
- Više konektora => ILI relacija

- Isto rešenje (zamena) cifre nezavisna od stanja u kome se nalazi ta cifra
- Kompaktnija predstava: AND-OR graf



AND-OR graf

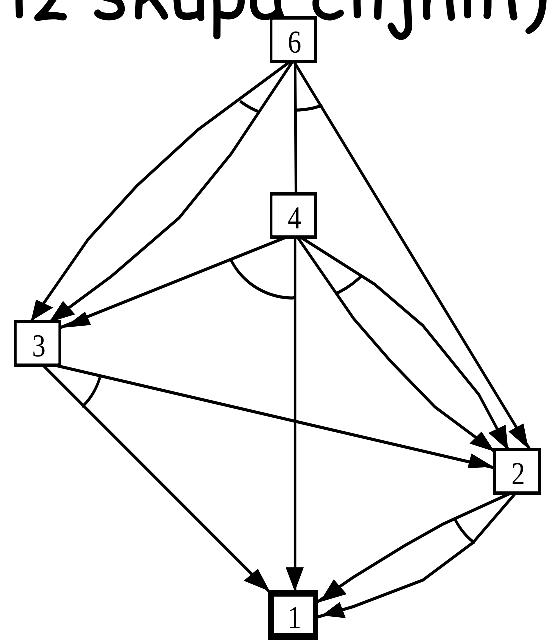


Kako se vrši pretraga AND-OR grafa ?

- Klasične metode pretrage: putanja od početnog do ciljnog čvora
- Kod AND-OR grafa cilj se predstavlja skupom ciljnih čvorova
- Rešenje je predstavljeno podgrafom G′ kompletnog grafa G

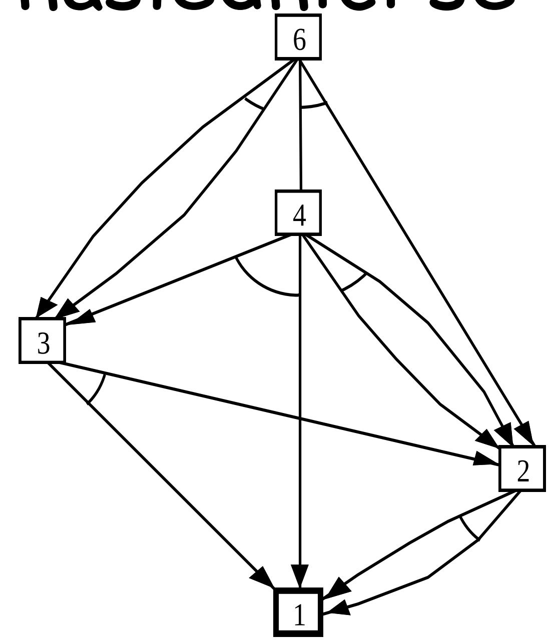
Kako se vrši pretraga AND-OR grafa ?

- Polazi se od startnog čvora
- Izabere se jedan od konektora
- Ukoliko svaki naslednik predstavlja ciljni čvor (jedan od čvorava iz skupa ciljnih) rešenje je pronadjeno



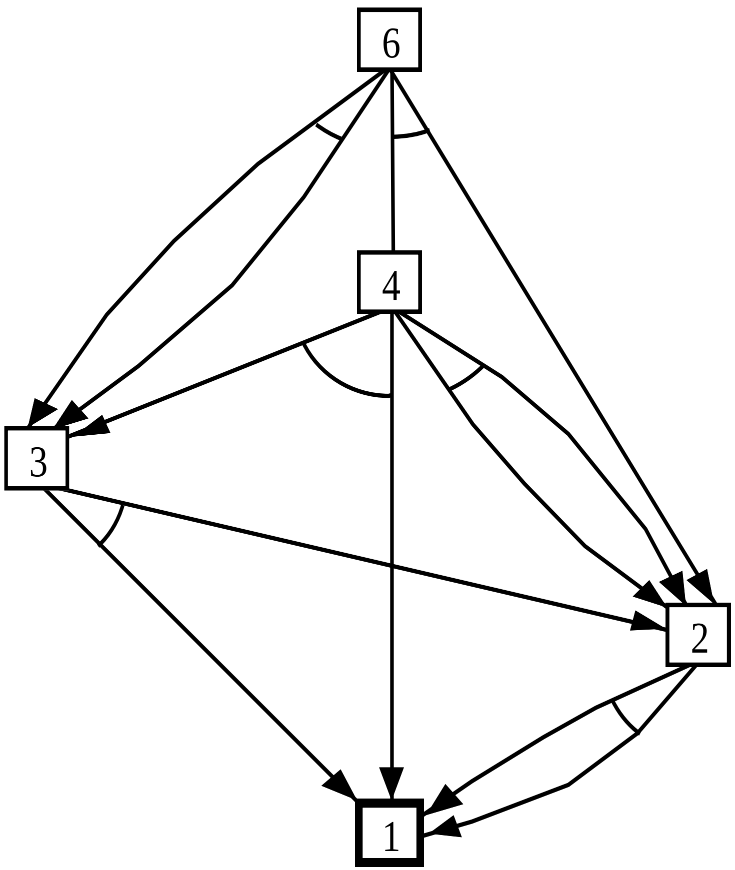
Kako se vrši pretraga AND-OR grafa ?

- U suprotnom za svaki od čvorova naslednika koji nije ciljni bira se jedan od konektora
- Izabrani konektori i naslednici se uključuju u rešenje

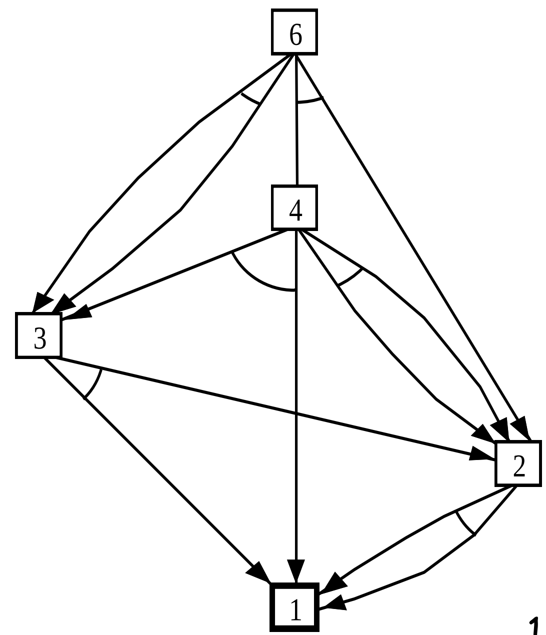


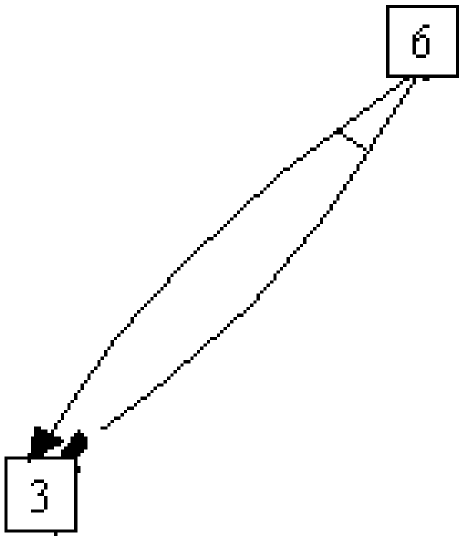
- Napomena: izbor konektora je proizvoljan
- => može postojati više rešenja

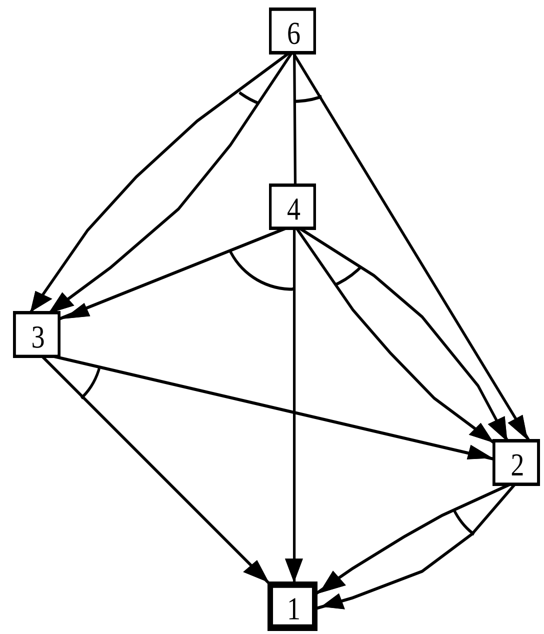
Izvršiti pretragu

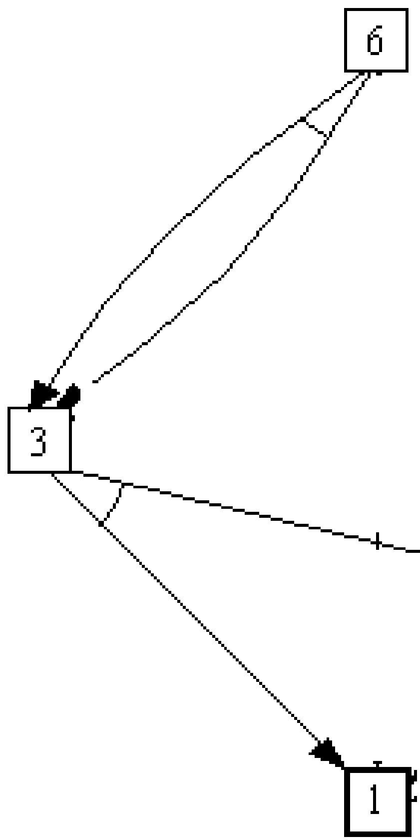


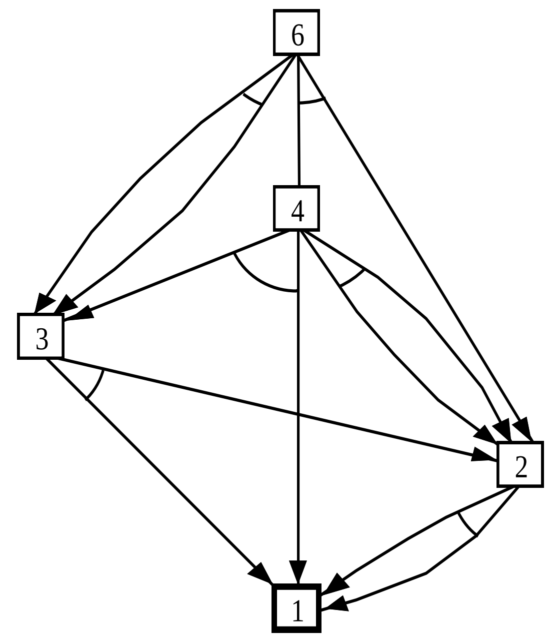
6

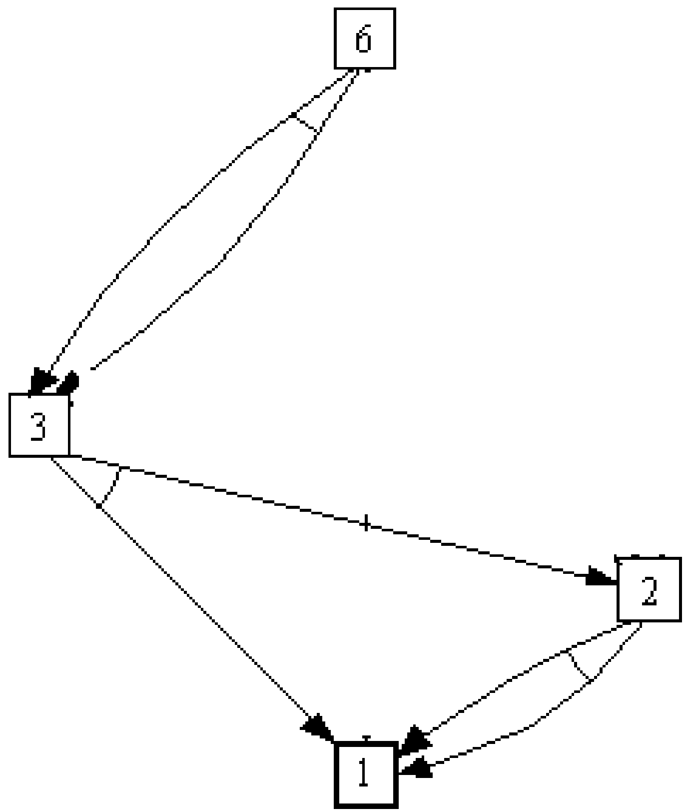


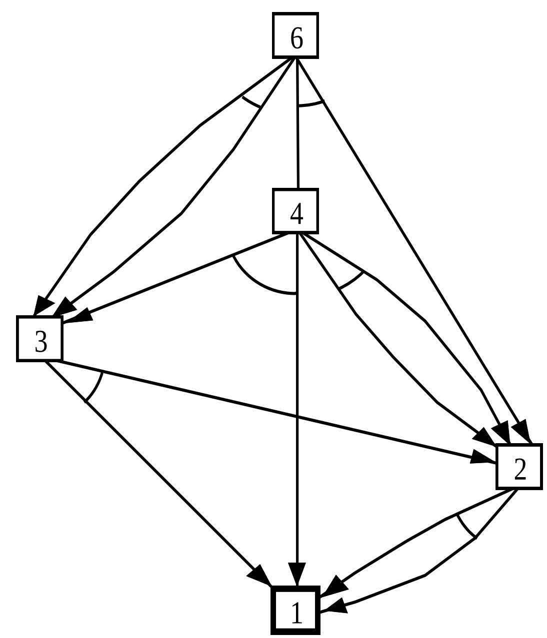

6
3


6
4
3
2
1

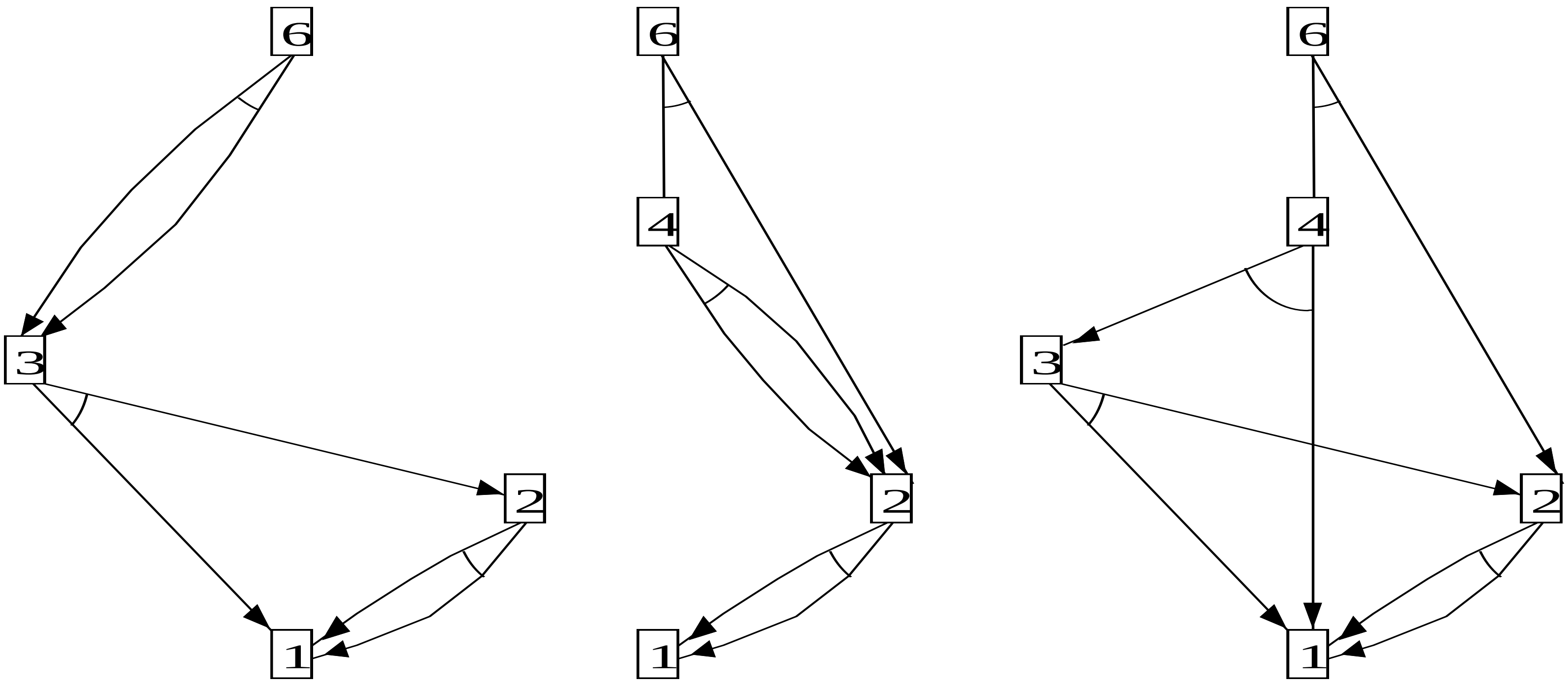

6
3
2
1


6
4
3
2
1


6
3
2
1


6
4
3
2
1

Moguća rešenja

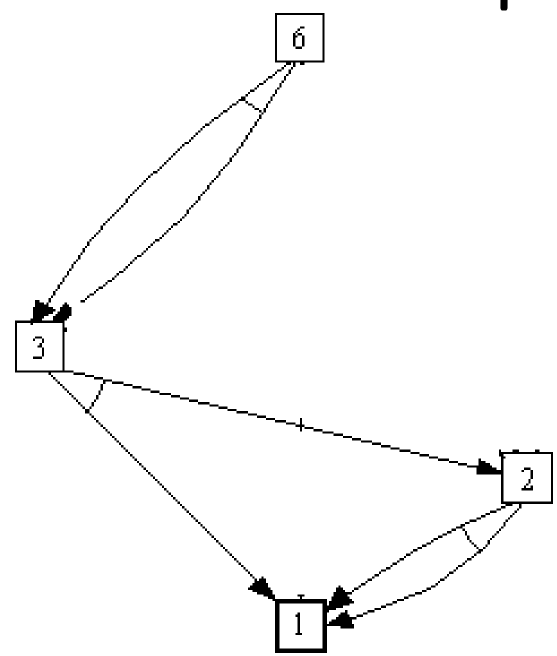


(a) (b) (c)

AND-OR + cena pravila

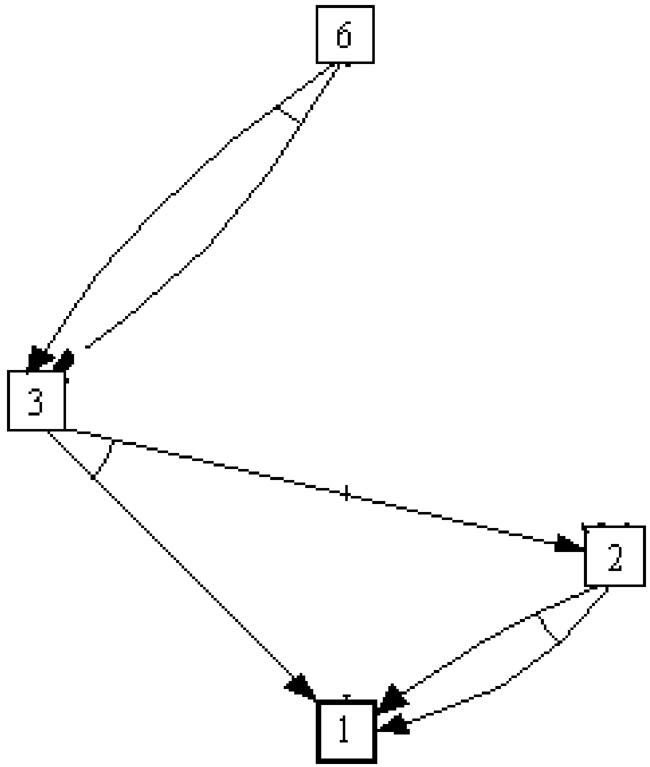
- Konektorima se mogu pridružiti cene, koje reprezentuju cene upotrebe pravila
- Na osnovu prethodnog može se definisati cena rešenja
- => možemo porediti rešenja

- Cena k(n,N) za neki podgraf G' grafa G
- n - startni čvor podgrafa
- N – skup ciljnih čvorova
- Ako je n element skupa N, onda je k(n,N) = 0
- Inače, čvor n poseduje konektor ka skupu čvorova n1, n2...

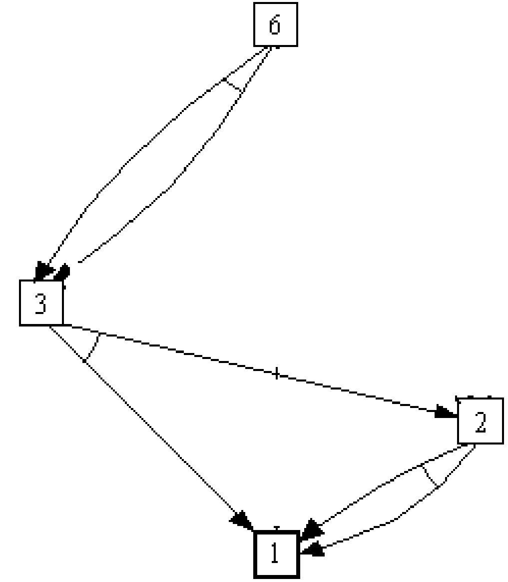


- Cena ovog konektora c
- Cena kompletnog rešenja

$k(n,N) = c + k(n1,N) + k(n2,N) + \ldots$



Naš primer



- N = {1}

- k(6,{1}) = k(P1) + k(3,{1}) + k(3,{1})
- k(3,{1}) = k(P5) + k(2,{1}) + k(1,{1})
- k(2,{1}) = k(P6) + k(1,{1}) + k(1,{1})
- k(1,{1}) = 0
- k(6,{1}) = 10

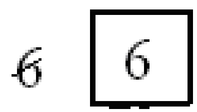
AO*

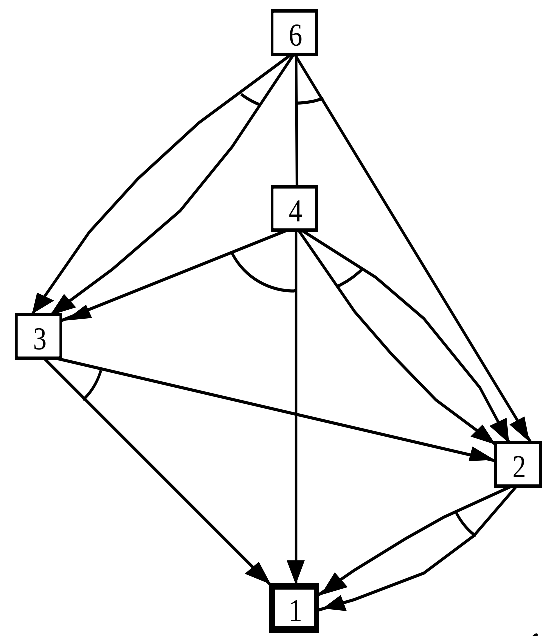
- Analogija A* algoritmu u slučaju pretrage AND-OR grafa
- Optimalno rešenje
- Heuristička funkcija za svaki čvor

AO*

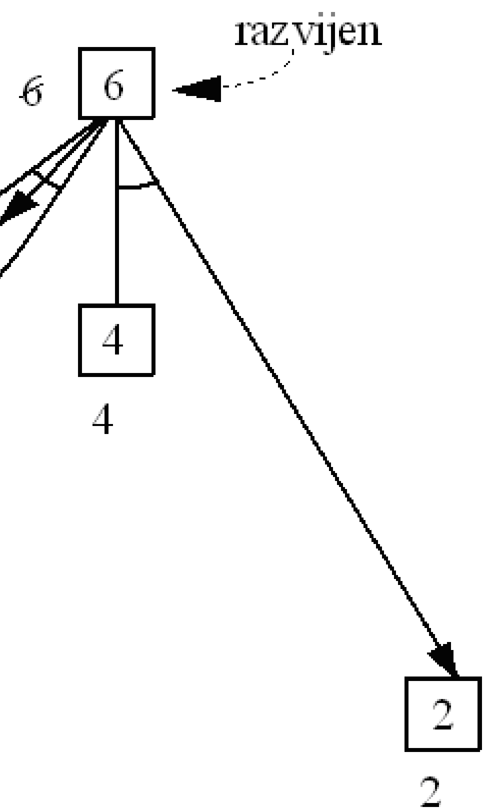
- Algoritam AO*
  - Ekspanzija izabranog čvora
  - Revizije funkcija procene čvorova grafa

- Inicijalno samo startni čvor
- Funkcija procene jednaka je njegovoj heurističkoj funkciji

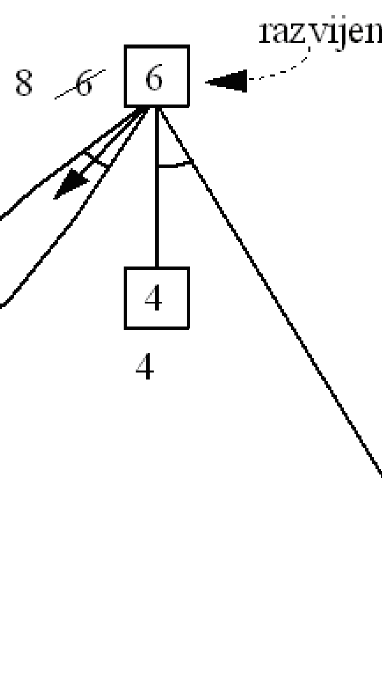


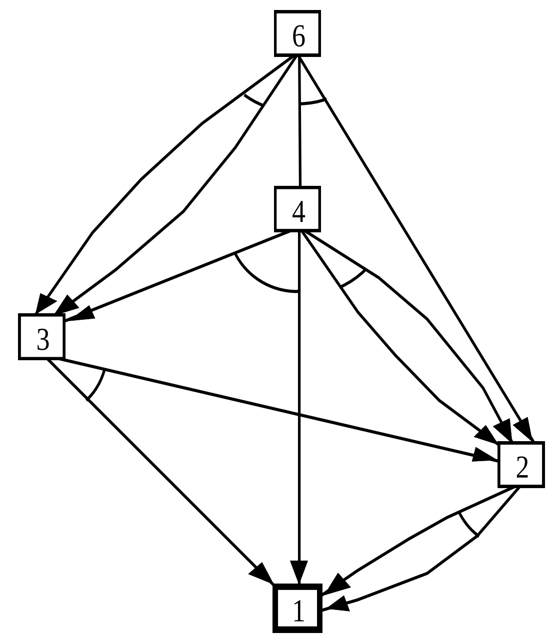


- Razvijamo čvor 6
- Unosimo konektore i čvorove 2, 3 i 4
- Funkcije procene (novih čvorova) jednake su heurističkim funkcijama

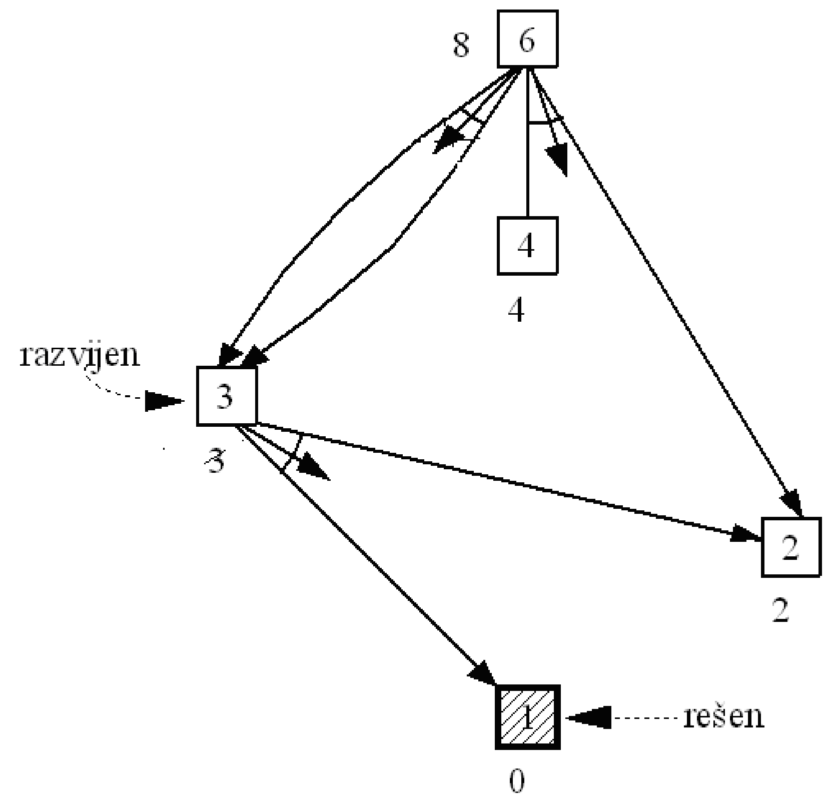


- Revidiranje funkcije procene čvora 6
- Po levom konektoru: $f = c + f(3) + f(3) = 2 + 3 + 3 = 8$
- Po desnom konektoru $f = c + f(4) + f(2) = 2 + 4 + 2 = 8$
- $f(6) = \min(f ,f) = 8$

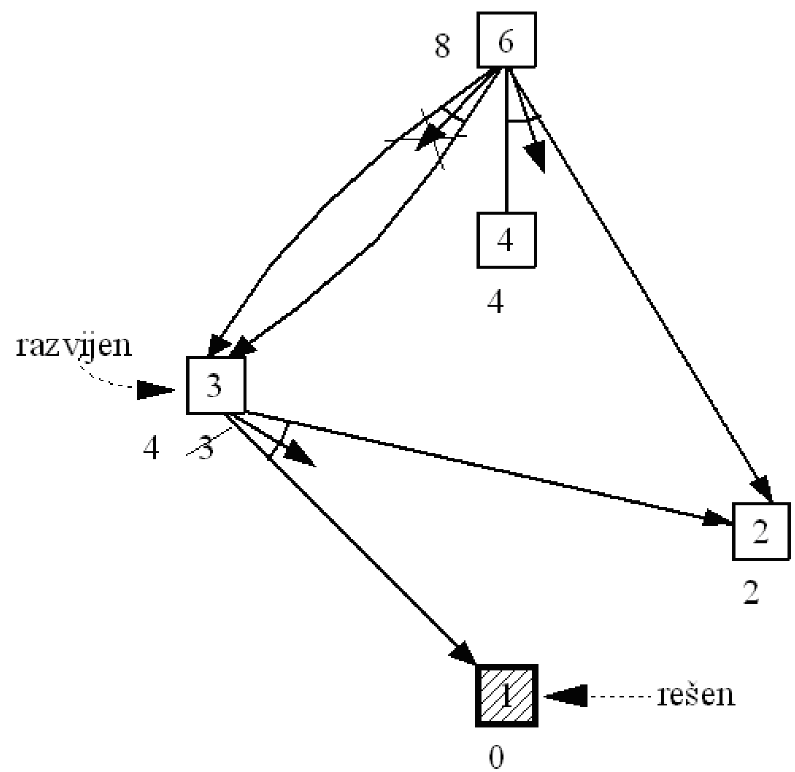




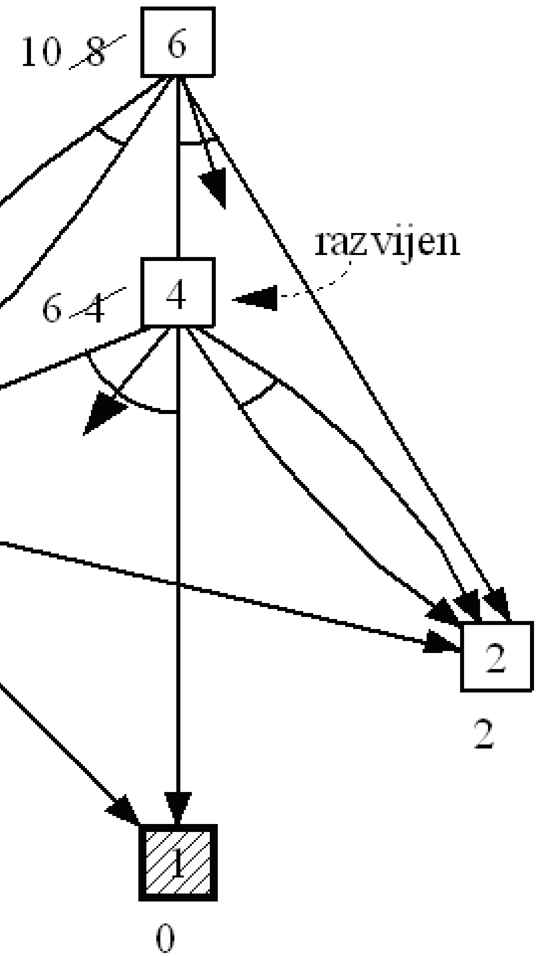
- Izbor novog čvora
- Čvor treba da se nalazi na najboljoj putanji
- Biramo čvor 3
- Čvor dva nije rešen

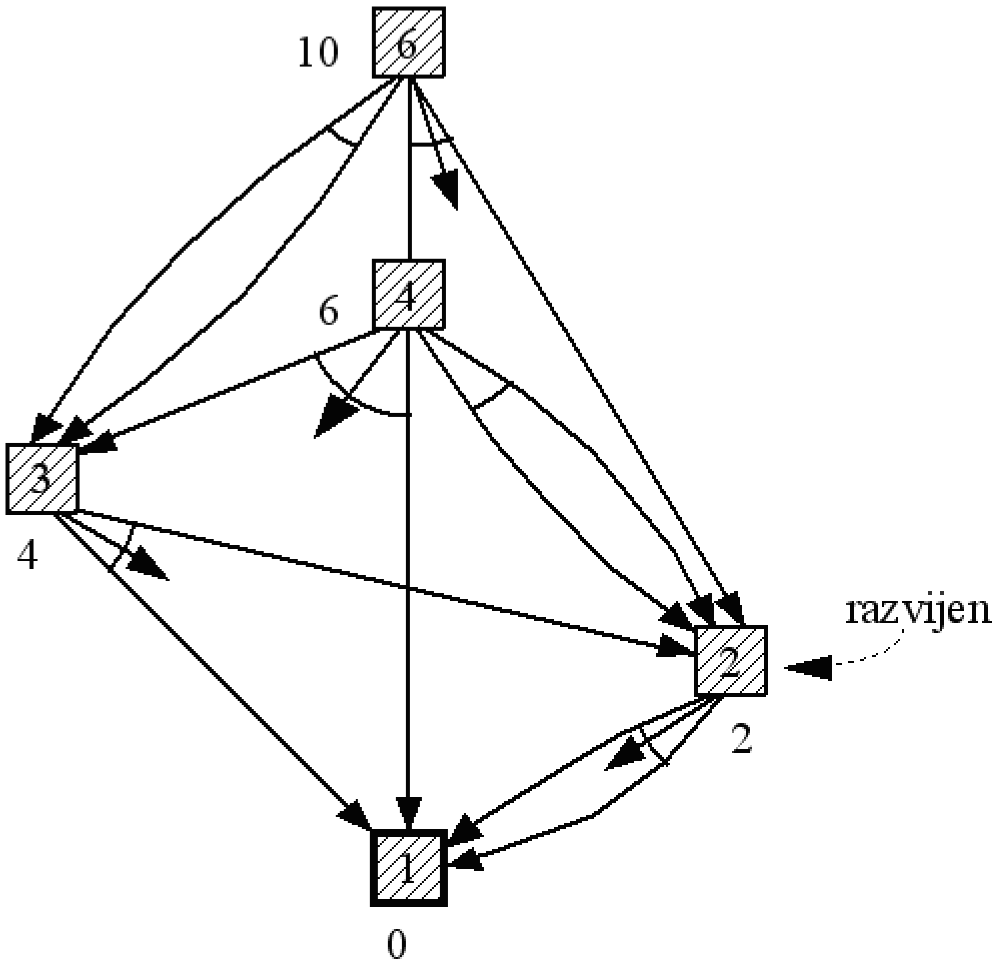


- Nova vrednost funkcije procene za čvor 3 jednaka je: f(3) = c + f(2) + f(1) = 2 + 2 + 0 = 4
- Potrebno je izvršiti promenu procene čvorova unazad (čvora 6)
- Po konektoru P1: f = c + f(3) + f(3) = 2 + 4 + 4 = 10
- f(6) = min(f , f) = 8



- Razvijamo čvor 4 (proizvoljno izabran)
- $f \quad = c \quad + f(2) + f(2) = 2 + 2 + 2 = 6$
- $f \quad = c \quad + f(3) + f(1) = 2 + 4 + 0 = 6$





(b)

Dodatno pogledati (stara zbirka)

- zadatak 9 (Viktorija)
- zadatak 10 (Putna Mreža)
- zadatak 19 (Igra nim)
- zadatak 20 (Problem šest kraljica)