

ALGORITMI NA GRAFOVIMA

PROBLEM

Po urbanističkom planu, dva grada su međusobno povezana određenim skupom kablova različite namene, međutim nisu svaka dva grada direktno povezana. Ako se dva grada ne mogu direktno povezati, njihova veza se ostvaruje preko grada koji jeste direktno povezan sa njima. Telekomunikaciona kompanija želi da zakupi neke od tih kablova kako bi počela da radi na tom tržištu, a kako cena zakupa zavisi od dužine kablova, želi da je zakup najmanje košta.

Problem se može predstaviti grafom gde čvorovi grafa predstavljaju gradove a kablovi grane koje dva čvora povezuju. Telekomunikaciona kompanija bi svoje sedište postavila u jednom gradu i do ostalih gradova došla iz svog sedišta po najkaćem putu. Odnosno, jezikom teorije grafova, želimo da napravimo minimalni razapinjući podgraf datog težinskog grafa sa n čvorova.

Pomenuti problem predstavlja problem najmanjeg razapinjućeg stabla (eng. Minimum Spanning Tree). U okviru ovog kursa biće predstavljena dva algoritma za rešavanje MST problema koji spadaju u proždrljive algoritme (eng. greedy algorithms)

- Kruskalov algoritam
- Primov algoritam

Postupak određivanja minimalnog razapinjućeg stabla može se posmatrati i na sledeći način:

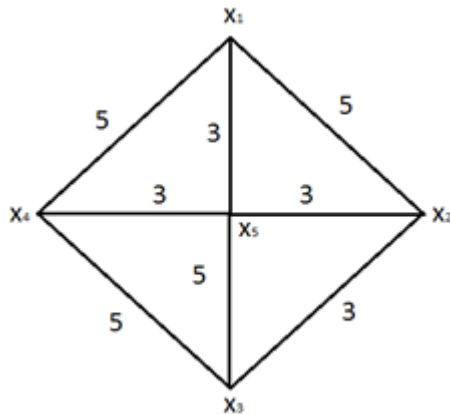
Neka je T razapinjuće stablo nad nekim grafom G i neka su T_1 i T_2 bilo koja dva poddrveća drveća T dobijena raskidanjem njihove trenutne veze. Posmatramo skup svih veza između T_1 i T_2 i povezujemo ih granom

$$\Delta_{ij} = \min_{x_i \in T_1} \{ \min_{x_j \in T_2} \{ c(x_i, x_j) \} \}.$$

Ponavljanjem ovog postupka može se dobiti najmanje razapinjuće stablo.

PRIMER 59:

Naći najmanje razapinjuće stablo za sledeći graf

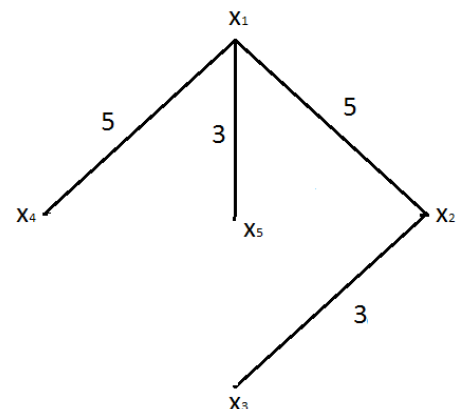


Rešenje

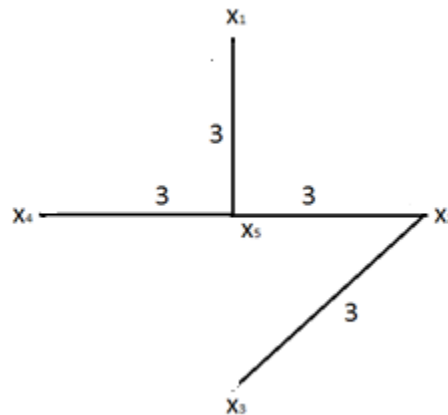
Prvo ćemo napraviti drvo čiji je koren npr. čvor x_1 .

Dobijeno drvo je težine 18. Naš zadatak je da napravimo drvo najmanje težine.

Za početak, ako razdvojimo čvor x_4 od drveća, odnosno prekinemo granu x_1x_4 primetićemo da će se jeftinija veza sa tim čvorom ostvariti preko čvora x_5 , odnosno preko grane x_5x_4 . Slično, veza između čvora x_2 i x_5 je jeftinija nego



preko nekog posrednog čvora pa ćemo i to zameniti. Konačno, najmanje razapinjuće stablo ima sledeći oblik:



Postoji više načina za dobijanje najmanjeg razapinjućeg stabla, mi ćemo posmatrati Kruskalov algoritam i Primov algoritam.

Kruskalov algoritam za nalaženje najmanjeg razapinjućeg stabla

Razapinjuće stablo minimalne težine u težinskom grafu G naziva se *optimalnim stablom*. Za konstrukciju optimalnog stabla koristimo tzv. Kruskalov algoritam koji se može opisati na sledeći način:

KORAK1 Početi od potpuno nepovezanog grafa T od n čvorova.

KORAK2 Poređati sve grane grafa G tako da im težina bude u rastućem redosledu

KORAK3 Počevši od najjeftinije grane dodavati ih grafu T vodeći računa da se ne napravi cikl.

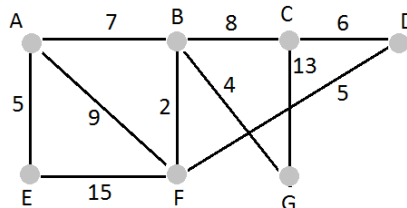
KORAK4 Ponoviti KORAK3 sve dok graf T ne bude imao $n - 1$ granu.

Dobijen podgraf T predstavlja minimalno razapinjuće stablo grafa G .

TEOREMA: Svako razapinjuće stablo konstruisano Kruskalovim algoritmom je optimalno.

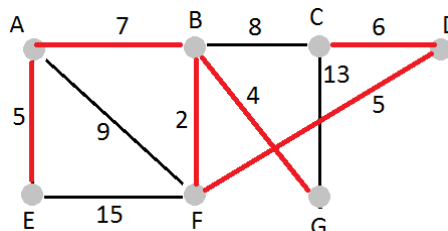
PRIMER 60

Primernom Kruskalovog algoritma odrediti minimalno razapinjuće stablo za sledeći graf:



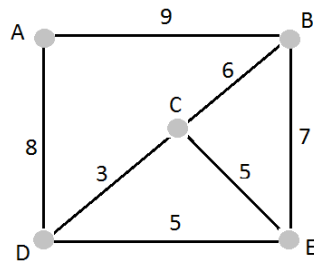
REŠENJE:

Odgovarajuće stablo je sledećeg oblika:



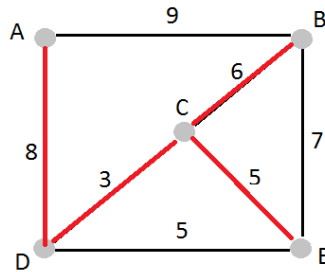
PRIMER 61

Primernom Kruskalovog algoritma odrediti minimalno razapinjuće stablo za sledeći graf:



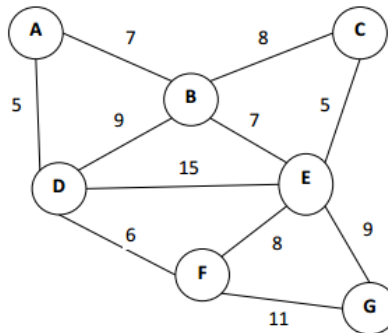
REŠENJE:

Odgovarajuće stablo je sledećeg oblika:



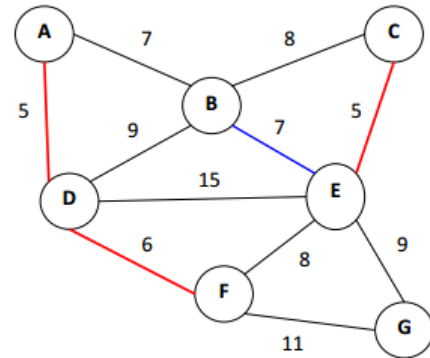
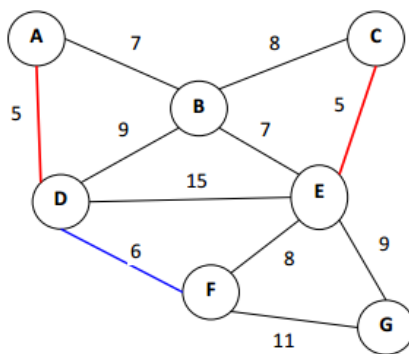
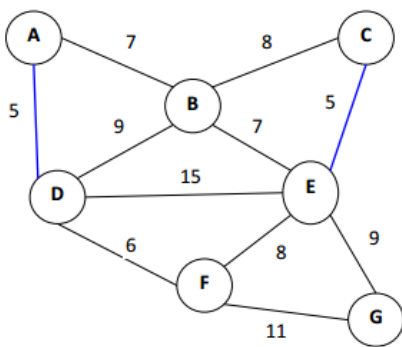
PRIMER 62

Primernom Kruskalovog algoritma odrediti minimalno razapinjuće stablo za graf dat slikom:

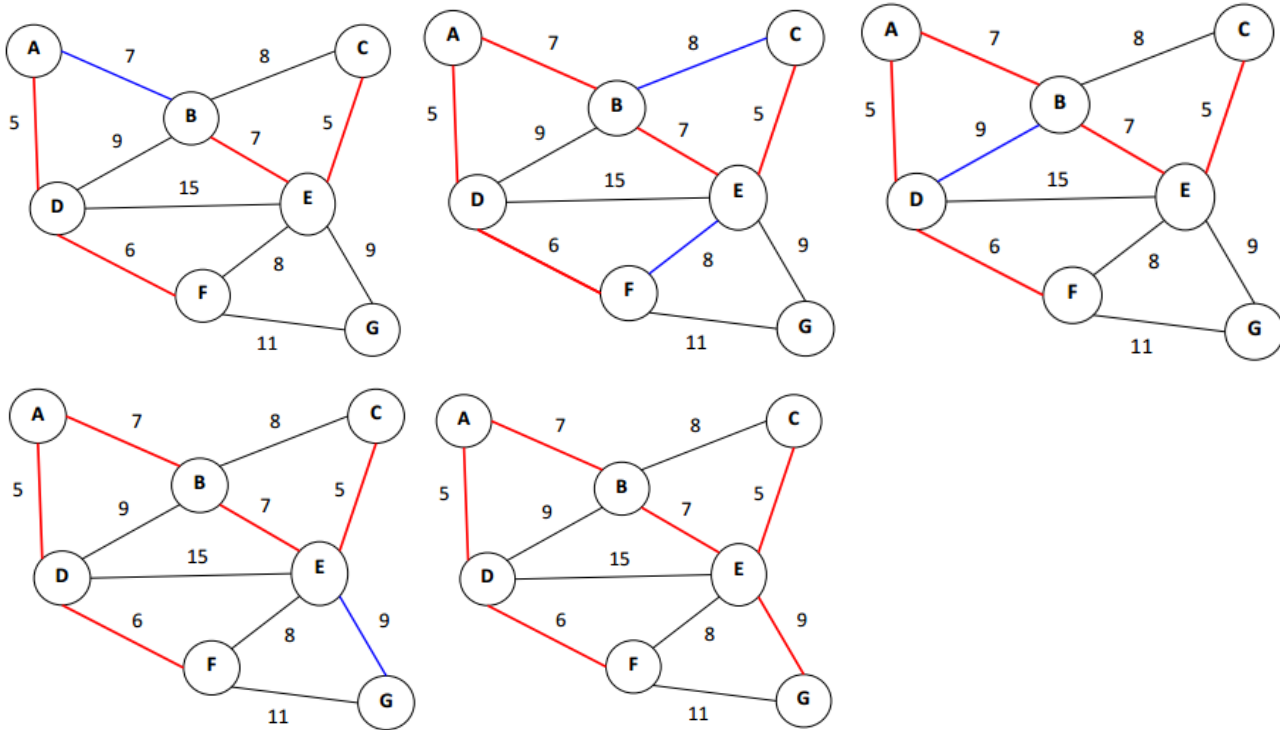


REŠENJE

Na početku, drvo T se sastoji se samo iz skupa čvorova $\{A, B, C, D, E, F, G\}$. Primetimo da su najjeftinije grane A-D i C-E, dodajmo ih drvetu (slika levo). Sledeće grane po veličini su DF (slika u sredini). Slede grane AB i BE (slika desno)



Sledeće grane po veličini su AB, BC i EF. Biramo granu AB s'tim da dodavanje grane BC nije dozvoljeno obzirom da bismo dobili cikl što nije dozvoljeno, tako da se ta grana ne može dalje uvrstiti u graf. Takođe, ni dodavanje grane EF nije dozvoljeno. Sledeća po veličini grana koju možemo je BD koju takođe ne možemo dodati. Dodajemo granu EG



Primov algoritam za nalaženje najmanjeg razapinjućeg stabla

Primov algoritam za nalaženje minimalnog razapinjućeg stabla može se konstruisati na sledeći način:

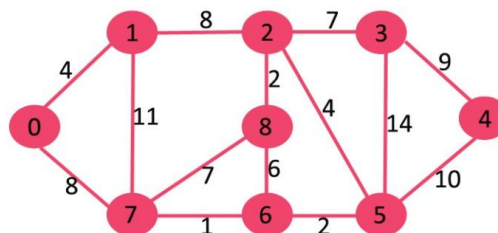
KORAK1 Proizvoljno uzeti nerazmatrani čvor (ako postoji) za tekuću komponentu. Na početku programa svi čvorovi se smatraju nerazmatranim. Ako takav čvor ne postoji preći na KORAK3 u protivnom izabrani čvor markirati kao razmazran.

KORAK2 Naći najlakšu granu koja povezuje čvor iz tekuće komponente sa nekim čvorom van nje. Ako takva grana ne postoji, preći na KORAK1. U protivnom, tu granu priključiti tekućoj komponenti kao i onaj krajnji čvor grane koji nije bio u komponenti. Ujedno markirati da je dotični čvor razmatran. Preći na početak KORAKA2.

KORAK3 Završiti algoritam.

PRIMER 63

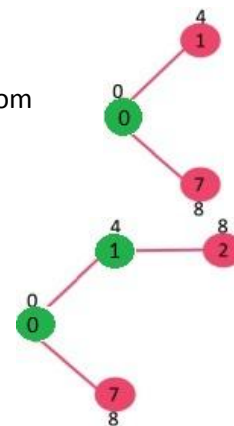
Odrediti minimalno razapinjuće stablo za sledeći graf



Rešenje

Obeležimo tekući skup sa $T = \{0, \infty, \infty, \dots, \infty\}$ gde ∞ predstavlja marker da veza sa odgovarajućim čvorom nije uspostavljena. Biramo čvor minimalnog rastojanja od čvora 0:

Na raspolaganju su nam čvorovi 1 i 7 čija odgovarajuća rastojanja iznose 4 i 8 (tim redom). Zelenom bojom obeležimo čvor koji smo markirali. Biramo čvor sa najmanjom težinom koji se do sada nije uključio u drvo T. Dakle, dodajemo čvor 1. Sada je drvo obliks $T = \{0,1\}$.

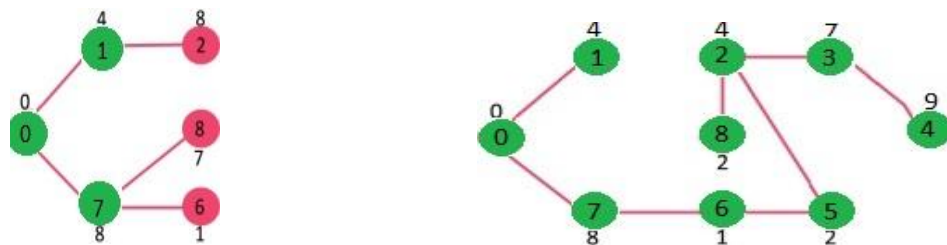


Čvor 1 povezan je čvorovima 2 i 7 od kojih je čvor 7 već jednom razmatran. Dakle, ili ćemo dodati čvor 7 koji je za 8 udaljen od početnog čvora ili čvor 2 koji j za 8 udaljen od čvora 1.

Novo stablo ima sledeći oblik $T = \{0,1,2\}$. Čvor 1 obeležimo kao već razmatrani čvor.

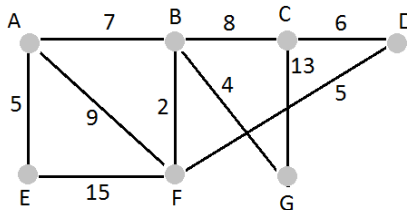
Dalje posmatramo odstojanje od čvorova 1 i 7. Primetimo da su nova rastojanja 8, 7 i 1. Proširujemo naše drvo dodavanjem čvora 6. $T = \{0, 1, 7, 6\}$.

Novi čvorovi za razmatranje su 2, 8 i 6 (na rastojanju 8, 6, 2). Biramo čvor 6 i tako redom. Konačno, traženo stablo je oblika:



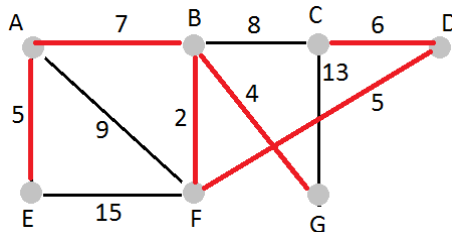
PRIMER 64

Primernom Primovog algoritma odrediti minimalno razapinjuće stablo za sledeći graf:



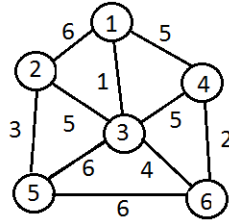
REŠENJE:

Odgovarajuće tablo je sledećeg oblika:



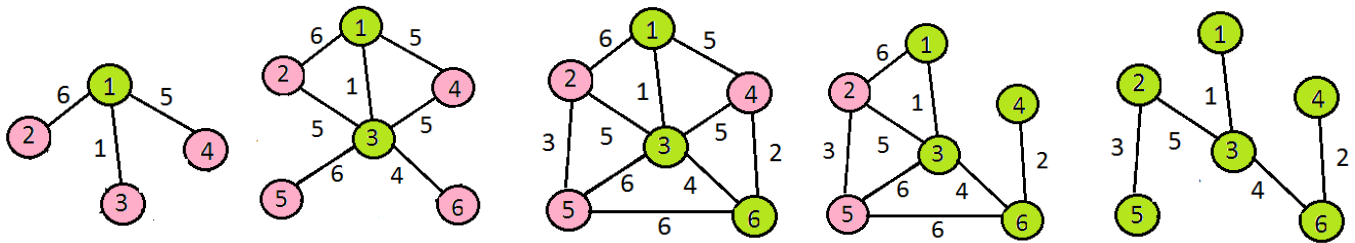
PRIMER 65

Odrediti minimalno razapinjuće stablo za sledeći graf



Rešenje:

Biramo čvor 1 kao početni. Rastojanje od čvora 1 do svih ostalih čvorova iznosi $\{6,1,5, \infty, \infty, \infty\}$. Biramo čvor 3 kao sledeći. Dakle $T = \{1,3\}$. Rastojanje stabla T od preostalih čvorova sada iznosi $\{5,5,6,4\}$. Biramo čvor 6 kao naredni...



Algoritam Dijkstre za nalaženje najkraćeg rastojanja između dva čvora

Algoritam Dijkstre je razvijen 1959. i smatra se jednim od najefikasnijih i najpogodnijih algoritama za određivanje najkraćeg puta između dva zadata čvora. Neophodan uslov za primenu ovog algoritma je da je dužina c_{ij} svake grane $(i, j) \in L$ nenegativna.

Prva tri koraka određuju dužinu najkraćeg puta dok poslednji korak identifikuje traženi put

KORAK 1

Svim čvorovima mreže dodeliti sledeće oznake: čvoru s dodeliti oznaku $d^+(s) = 0$ koju ćemo smatrati nepromenljivom, a svim ostalim čvorovima, uključujući i čvor t , dodeliti promenljive oznake $d^-(i) = -\infty$ a zatim preći na KORAK2.

KORAK 2

Staviti $s = i$, a zatim odrediti skup čvorova $j \in A_i$ sa promenljivim oznakama gde A_i predstavlja skup čvorova sledbenika čvora i . Za takve čvorove odrediti nove, takođe promenljive oznake na sledeći način:

$$d(j) = \min \{d^-(j), d(i) + c_{ij}\} \quad (1)$$

Od svih novodobijenih oznaka samo jedna oznaka je nepromenljiva i određuje se iz uslova

$$d^+(j) = \min \{d(j)\} \quad (2)$$

Dok su sve ostale promenljive. Nakon određivanja ovih oznaka preći na KORAK 3.

KORAK 3

Proveriti da li je čvor t označen nepromenljivom oznakom i u zavisnosti od toga postoje sledeće dve mogućnosti:

K 3.1: Čvor t je označen nepromenljivom oznakom $d^+(t)$, koja u stvari predstavlja dužinu traženog najkraćeg puta, pa se postupak obeležavanja čvorova završava i potrebno je preći na postupak izdvajanja najkraćeg pita, odnosno preći na korak4.

K 3.2: Čvor t nije označen nepromenljivom oznakom. Staviti da je $i = j$, gde je j čvor koji u prethodnom postupku dobio nepromenljivu oznaku $d^+(j)$ i vratiti se na korak3.

KORAK 4

Ovim korakom se izdvaja najkraći put. Postupak počinjemo iz čvora t . U poslednjoj iteraciji čvoru t dodeljena je nepromenljiva oznaka $d^+(t)$ dok su u prethodnim iteracijama preostalim čvorovima mreže $i \in N$ dodeljene ili promenljive oznake $d^-(i)$ ili nepromenljive oznake $d^+(i)$. Od svih čvorova susednih čvoru t odredimo čvor j_k sa nepromenljivom oznakom $d^+(j_k)$ za koji je

$$d^+(t) - d^+(j_k) = c_{j_k t} \quad (3)$$

Vidimo da čvor j_k sigurno postoji, jer se pomoću takvog čvora i određuje nepromenljiva oznaka čvora t . Isto tako je jasno da za sve ostale čvorove susedne čvoru t , koje možemo označiti sa q_r i koji mogu imati promenljive ili nepromenljive oznake, važi

$$d^+(t) - d(q_r) \leq c_{q_r t} \quad (4)$$

Ponavljanjem ovog postupka, to jest određivanjem čvora j_{k-1} , susednog čvoru j_k , koji ima nepromenljivu oznaku $d^+(j_{k-1})$, za koji je

$$d^+(j_k) - d^+(j_{k-1}) = c_{j_{k-1}, j_k} \quad (5)$$

Na kraju se dolazi do čvora j_1 , kome je susedni čvor s , i za koji je

$$d^+(j_1) - d^+(s) = c_{s j_1} \quad (6)$$

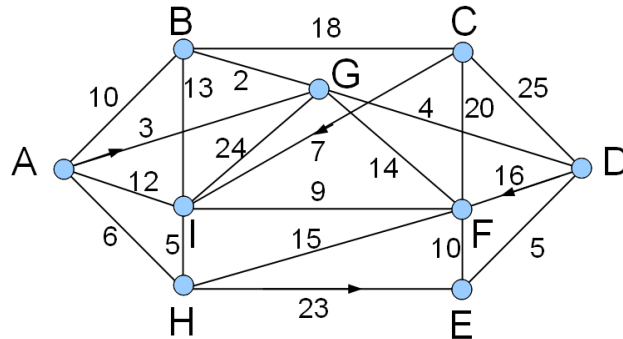
Sada smo dobili put koji ide preko čvorova $s, j_1, j_2, \dots, j_{k-1}, j_k, t$. Sabiranjem relacija (3), (5) i (6) za sve čvorove na tom putu (vodimo računa o tome da je $d(s) = 0$) dobijamo da je dužina puta jednaka nepromenljivoj oznaci čvora t , to jest dobijamo da je dužina puta

$$d^+(t) = c_{sj_1} + c_{j_1j_2} + \dots + c_{j_k t} \quad (7)$$

Dobijeni put je i najkraći put između zadatih čvorova s i t .

PRIMER 66:

Koristeći algoritam Dijkstra-e naći najkraći put između čvorova A i bilo kog drugog čvora date mreže.



- Mreža može imati više najkraćih puteva između dva čvora i to se može uočiti u toku same postavke algoritma (postoji više čvorova koji zadovoljavaju relaciju 2).

Modifikacija algoritma Dijkstra-e

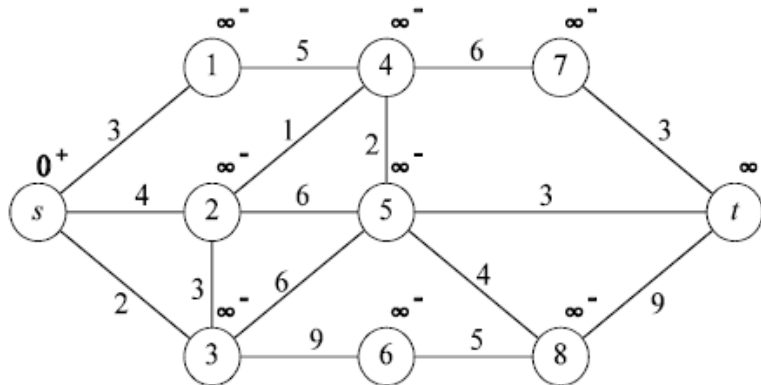
- Prva dva koraka se spoje i prilikom određivanja oznaka čvorova pored svake nepromenljive oznake se upisuje i indeks čvora na osnovu kojeg je, u prethodnom koraku, dobijena ta oznaka, onda se time može i rešiti i problem izdvajanja najkraćeg puta. Dakle, svaki čvor $j \in N$ i $j \neq s$ umesto ranije nepromenljive oznake $d^+(j)$ sada bi imao isto tako nepromenljivu, oznaku koja se, po konvenciji, može pisati na jedan od sledećih načina:

$$[i, d^+(j)] \text{ ili } [d^+(j), i] \text{ ili } [i, d(j)] \text{ ili } [d(j), i].$$

gde je i čvor susjedni čvoru j pomoću kojeg se, primenom relacije (7) određuje oznaka $d^+(j)$. Na ovaj način bi čvor t dobio novu oznaku, recimo $[d(t), j_k]$, što znači da najkraći put do čvora t ide preko čvora j_k . Čvor j_k je već ranije označen novom oznakom $[d(j_k), j_{k-1}]$, što znači da najkraći put ide i preko čvora j_{k-1} . Produžavajući ovaj postupak na kraju se dolazi do čvora s . Ovim postupkom se problem izdvajanja najkraćeg puta jednostavno rešava, bez ikakvih dopunskih izračunavanja.

PRIMER 67

Koristeći Dijkstrin algoritam odrediti najkraće rastojanje od čvora s do čvora t .



Rešenje:

Zadata mreža sa početnim oznakama je data na slici. Prelazimo na iterativni deo primene obeležja. Sitna obeležja su data na slici.

- 1) Čvor s slede čvorovi 1,2,3 i $A_s = \{1,2,3\}$

$$d^-(1) = \min\{0 + 3, \infty\} = 3$$

$$d^-(2) = \min\{0 + 4, \infty\} = 4$$

$$d^-(3) = \min\{0 + 2, \infty\} = 2$$

$$\min\{d^-(1), d^-(2), d^-(3)\} = d^-(3) = 2 \rightarrow d^+(3) = d^-(3)$$

- 2) Čvor 3 slede čvorovi 2, 5, 6 i $A_3 = \{2,5,6\}$

$$d^-(2) = \min\{2 + 3, 4\} = 4$$

$$d^-(5) = \min\{2 + 6, \infty\} = 8$$

$$d^-(6) = \min\{2 + 9, \infty\} = 11$$

$$\min\{d^-(1), d^-(2), d^-(5), d^-(6)\} = d^-(1) = 3 \rightarrow d^+(1) = d^-(1)$$

- 3) $A_1 = \{4\}$

$$d^-(4) = \min\{3 + 5, \infty\} = 8$$

$$\min\{d^-(2), d^-(4), d^-(5), d^-(6)\} = d^-(2) = 4 \rightarrow d^+(2) = d^-(2)$$

- 4) $A_2 = \{4,5\}$

$$d^-(4) = \min\{4 + 1, 8\} = 5$$

$$d^-(5) = \min\{4 + 6, 8\} = 8$$

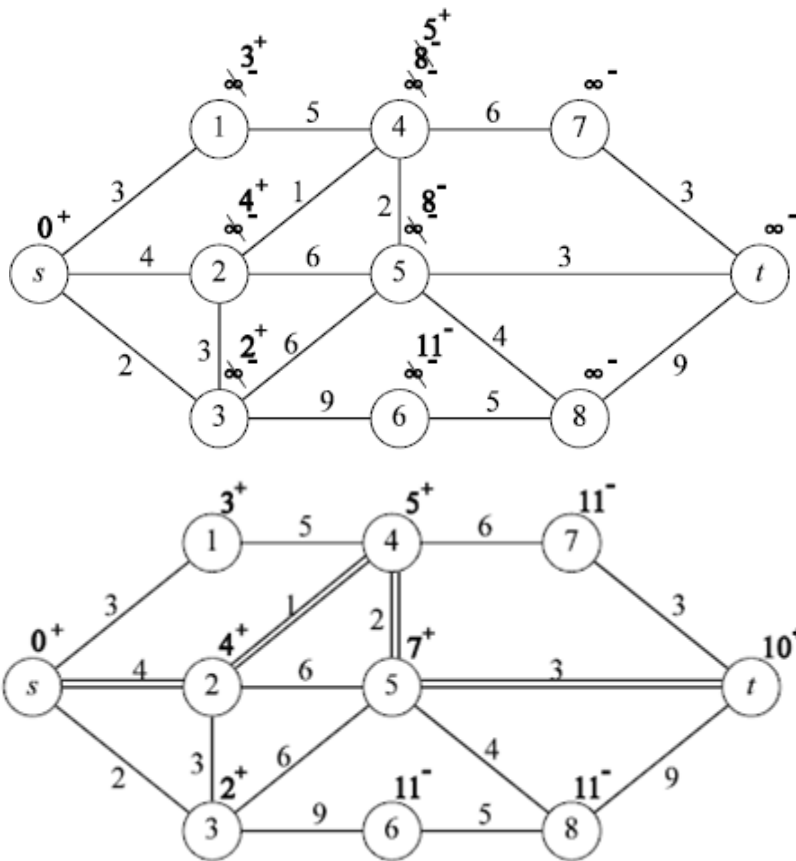
$$\min\{d^-(4), d^-(5), d^-(6)\} = d^-(4) = 5 \rightarrow d^+(4) = d^-(4)$$

- 5) $A_2 = \{5,7\}$

$$d^-(5) = \min\{5 + 2, 8\} = 7$$

$$d^-(7) = \min\{5 + 6, \infty\} = 11$$

$$\min\{d^-(5), d^-(6), d^-(7)\} = d^-(5) = 7 \rightarrow d^+(5) = d^-(5)$$



Prelazimo na određivanje najkraćeg puta:

$$10 - 3 = 7 \rightarrow \text{čvor } 5$$

$$7 - 2 = 5 \rightarrow \text{čvor } 4$$

$$5 - 1 = 4 \rightarrow \text{čvor } 2 \text{ i}$$

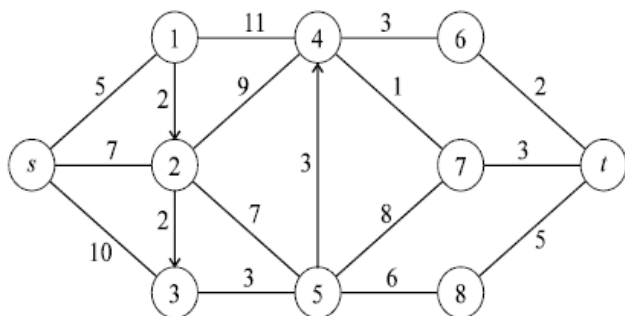
$$4 - 4 = 0 \rightarrow \text{vratismo se na početni čvor } s.$$

Najkraći put u mreži je $(s, 2, 3, 4, 5, t)$

Korisno je još proveriti da li dužina dobijenog puta odgovara obeležju $d^+(t)$: $3 + 2 + 1 + 4 = 10$.

Potrebno je napomenuti da stalno obeležje čvora predstavlja dužinu najkaćeg puta od čvora s do tog čvora. Kada se rešava problem određivanja najkraćeg puta između početnog i svih ostalih čvorova u mreži, potrebno je produžiti postupak obeležavanja dok svi čvorovi ne dobiju stalna obeležja.

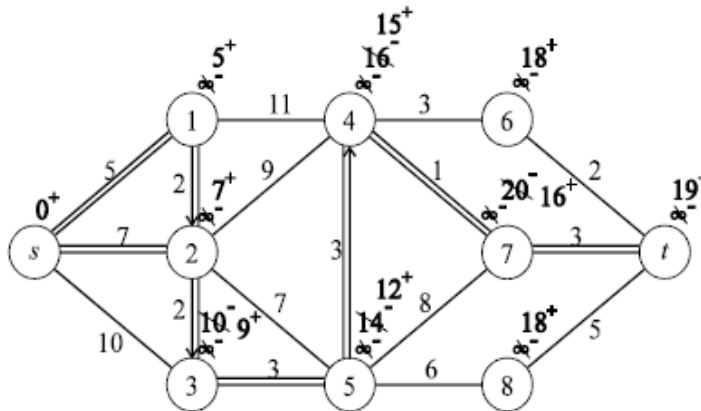
PRIMER 68:



- Odrediti najkraći put između čvorova s i t i njegovu dužinu koristeći Dijkstrin algoritam
- Odrediti dužine najkraćih puteva između čvora s i svih preostalih čvorova u mreži

Rešenje

Zadatak ćemo rešavati na samoj mreži:



a) Dužina najkraćeg puta je 19. Kod određivanja puta otkrivamo da je rešenje višestruko jer u čvoru 2 postoje dva prethodna čvora koja zadovoljavaju uslov $d^+(2) - d^+(j) = c_{j2}$ gde j predstavlja susedne čvorove čvora 2. Ti putevi su:

I: $(s, 1, 2, 3, 5, 4, 7, t)$

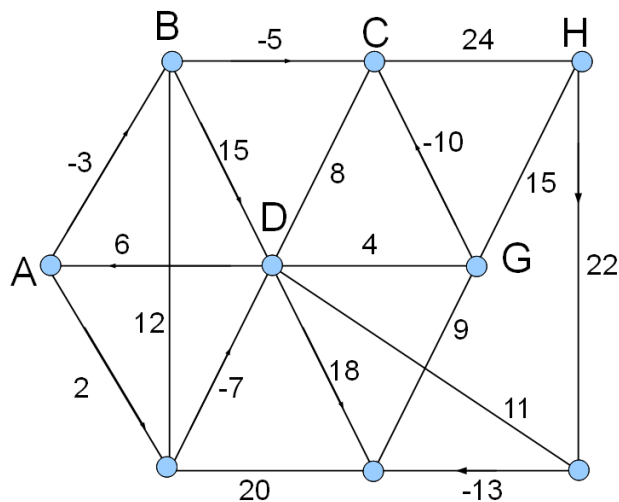
II: $(s, 2, 3, 5, 4, 7, t)$

b) Ovaj deo zadatka je implicitno već rešen u delu pod a).

Vidimo da su svi čvorovi na mreži obeleženi stalnim obeležjima koja predstavljaju dužine najkraćih puteva od čvora s do njih. Dakle, odgovarajuće dužine iznose:

$s - 1$: 5	$s - 5$: 12	$s - t$: 19
$s - 2$: 7	$s - 6$: 18		
$s - 3$: 9	$s - 7$: 16		
$s - 4$: 15	$s - 8$: 18		

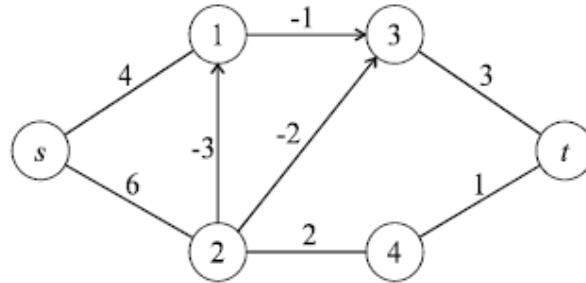
PRIMER 69:



Primer je uzet iz knjige Graph theory.

PRIMER 70:

Odrediti najkraći put i njegovu dužinu za čvorove s i t mreže prikazane na slici



Rešenje:

$$\begin{aligned}
 k = 1, R = \{1, 2\} \\
 d^1(s) = 0 & \qquad d^1(3) = \infty \\
 d^1(1) = 4 & \qquad d^1(4) = \infty \\
 d^1(2) = 6 & \qquad d^1(t) = \infty
 \end{aligned}$$

Čvorovi koji slede skup čvorova R su: $\Gamma(R) = \{s, 1, 3, 4\}$ i za njih računamo nove oznake:

$$d^2(s) = \min\{0, \min[4+4, 6+6]\} = 0$$

$$d^2(1) = \min\{4, \min[0+4, 6-3]\} = \min\{4, 3\} = 3$$

$$d^2(3) = \min\{\infty, 4-1, 6-2\} = 3$$

$$d^2(4) = \min\{\infty, 6+2\} = 8$$

Pošto je došlo do promene bar jednog $d^{k-1}(i)$ (označeni sa „ \rightarrow “), nastavljamo sa iteracijama.

$$\begin{aligned}
 k = 2, R = \{1, 3, 4\} \\
 \Gamma(R) = \{s, 2, 3, t\} \\
 d^3(s) = \min\{0, 4+4, 6+6\} = 0 \\
 d^3(2) = \min\{6, 8+2\} = 6 \\
 d^3(3) = \min\{3, 6-2, 3-1\} = 2 \\
 d^3(t) = \min\{\infty, 3+3, 8+1\} = 6
 \end{aligned}$$

$$\begin{aligned}
 k = 3, R = \{3, t\} \\
 \Gamma(R) = \{3, 4, t\} \\
 d^4(3) = \min\{2, 6-2, 3-1, 6+3\} = 2 \\
 d^4(4) = \min\{8, 6+2, 6+1\} = 7 \\
 d^4(t) = \min\{6, 2+3, 8+1\} = 5
 \end{aligned}$$

$$\begin{aligned}
 k = 4, R = \{4, t\} \\
 i \in \{2, 3, 4, t\} \\
 d^5(2) = \min\{6, 7+2\} = 6 \\
 d^5(3) = \min\{2, 6-2, 3-1, 5+3\} = 2 \\
 d^5(4) = \min\{7, 6+2, 5+1\} = 6 \\
 d^5(t) = \min\{5, 2+3, 8+1\} = 5
 \end{aligned}$$

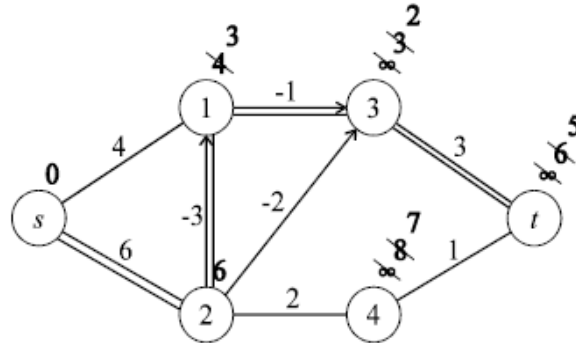
$$k = 5, R = \{4\}$$

$$i \in \{2, t\}$$

$$d^6(2) = \min\{6, 7+2\} = 6$$

$$d^6(t) = \min\{5, 2+3, 7+1\} = 5$$

Pošto u poslednjoj iteraciji nije promenjeno ni jedno obeležje a k je manje od $n = 6$, zaključujemo da je postupak završen. Dužina najkraćeg puta od čvora s do čvora t iznos 5 a put je $(s, 2, 1, 3, t)$. Poslednje obeležje čvora predstavlja dužinu najkraćeg puta od čvora s do tog čvora. Postupak određivanja puta i njegove dužine može se sprovesti i na samoj mreži:



Određivanje najkraćih puteva između svaka dva čvora date mreže

- Problem određivanja najkraćeg puta između svaka dva čvora date mreže može da se reši primenom algoritma Dijkstra-e ili primenom Bellmanovog algoritma kroz n ponavljanja (n – broj čvorova mreže) ali bi tada broj računskih operacija bio veliki. Iz ovog razloga razvili su se neki drugi algoritmi, jedan od njih je Floyd-ov algoritam.
 - Primenom Floyd-ovog algoritma mogu se odrediti samo dužine najkraćih puteva između svih parova čvorova date mreže ali se ne dobijaju informacije o identifikaciji puteva. Dakle, dobija se dužina najkraćeg puta ali ne i sam put.
 - Modifikacijom Floyd-ovog algoritma dobijamo Floyd-Warshall-ov algoritam pomoću koga se pored određivanja dužine najkraćeg puta identifikuje i sam put.
- Koristimo sledeće oznake:
- Dužine puteva između dva čvora su realni brojevi $c_{ij}, (i, j) \in L$
 - $c_{ij} = 0, i \in N$
 - $c_{ij} = \infty, (i, j) \notin L$

FLOYD-OV ALGORITAM ZA NALAŽENJE NAJKRAĆEG RASTOJANJA IZMEĐU SVAKA DVA ČVORA

KORAK 1

Posmatramo matricu $C^0 = [c_{ij}^0]$ na sledeći način:

$$c_{ij}^0 = c_{ij}, (i, j) \in N$$

$$c_{ij} = \infty, (i, j) \notin L$$

$$c_{ij} = 0, i \in N$$

Na osnovu matrice $C^0 = [c_{ij}^0]$ formiramo zatim matricu $D^0 = [d_{ij}^0]$ čiji su elementi

$$d_{ij}^0 = j \text{ ako je } c_{ij}^0 \neq \infty$$

$$d_{ij}^0 = 0 \text{ ako je } c_{ij}^0 = \infty$$

$$d_{ij} = 0, \quad i \in N$$

Preći na KORAK2.

KORAK 2

Staviti da je $k = 1$ (u opštem slučaju $k = k + 1$) i odrediti skupove čvorova I i J na sledeći način:

$$I = \{i | i \neq k, c_{ij}^0 \neq \infty\}$$

$$J = \{j | j \neq k, c_{kj}^0 \neq \infty\}$$

a zatim za svako $i \in I$ i $j \in J$ izračunati elemente matrice C^{k+1} pomoću sledeće relacije:

$$c_{ij}^{k+1} = \min \{c_{ij}^k, c_{i,k+1}^k + c_{k+1,j}^k\}$$

dok su svi preostali elementi matrice C^{k+1} jednaki odgovarajućim elementima matrice C^k .

Na osnovu matrica C^{k+1} i C^k određuju se i elementi matrice D^{k+1} na sledeći način

$$d_{ij}^{k+1} = k + 1 \text{ ako je } c_{ij}^{k+1} \neq c_{ij}^k$$

$$d_{ij}^{k+1} = d_{ij}^k \text{ ako je } c_{ij}^{k+1} = c_{ij}^k$$

Postupak određivanja matrica $C^{k+1} = [c_{ij}^{k+1}]$ i $D^{k+1} = [d_{ij}^{k+1}]$ se završava. Prelazi se na KORAK 3.

KORAK 3

Proveriti da li je $k \leq n$ a zatim preći na korake K 3.1-K 3.3.

K 3.1: Ako se za neko $k < n$ i ako su svi $c_{1i}^k \leq 0$ staviti da je $k = k + 1$ i ići na korak 2.

K 3.2: Ako se za bilo koje $k \leq n$ dobije da je $c_{ii}^k < 0$, to znači da postoji kontura negativne dužine koja sadrži čvor i . Ako se ovaj slučaj dogodi sledi da rešenje ne postoji i algoritam se završava.

K 3.3: Ako se ustanovi da je $k = n$ i da su svi $c_{ij}^n > 0$, tada se proces izračunavanja i određivanja elemenata matrice C^k i D^k završava, a elementi c_{ij}^n , matrice C^n predstavljaju tražene dužine najkraćih puteva između svaka dva čvora date mreže, dok se na osnovu elemenata d_{ij}^n matrice D^n vrši identifikacija tih puteva pto se obavlja u koraku br4.

KORAK 4

U ovom koraku se identifikuje najkraće puteve između čvorova i i j čije su dužine date u matrici C^n :

U posebnu tabelu upišemo broj jednak indeksu j a zatim uočimo element d_{ij}^n matrice D^n . Neka je taj element jednak nekom broju s_1 . Ovaj broj upišemo u tabelu ispred broja jednakog indeksu j , a zatim proveravamo da li je broj s_1 jednak indeksu i . U zavisnosti od toga imamo dve mogućnosti:

K 4.1: Ako je $s_1 = i$ tada je postupak identifikacije najkraćeg puta između čvorova i i j završen, a čvorovi preko kojih ide taj put su dati u tabeli i time se algoritam završava.

K 4.2: Ako je $s_1 \neq i$ onda uočimo element $d_{1s_1}^n$, matrice D^n , i neka je taj element jednak nekom broju s_2 a zatim proveravamo da li je taj broj jednak indeksu i ili nekom od ranije upisanih brojeva u tabeli. U zavisnosti od ove provere razlikujemo sledeće mogućnosti:

K 4.2.1: Ako je $s_2 = i$ onda se s_2 upisuje u tabelu ispred broja i i dalje se posupa kao u slučaju opisanom u kopraku K4.1.

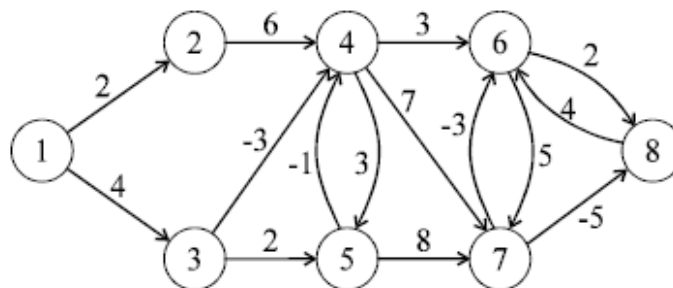
K 4.2.2: Ako je s_2 jednak bilo kome od ranije upisanih brojeva u tabeli, onda se u tabelu ispred broja s_1 upisuje broj jednak indeksu i , a dalje identifikacija najkraćeg puta je ista kao u koraku K4.1.

K 4.2.3: Ako je broj s_2 različit i od indeksa i i od svih brojeva ranije upisanih u tabelu, onda uočimo element $d_{1s_2}^n$ matrice D^n . Neka je taj element jednak broju s_3 . Proverimo da li je broj s_3 jednak indeksu i ili nekom od ranije upisanih brojeva i u zavisnosti od te provere koristi se neka od mogućnosti opisana u koracima K 4.2.1, K4.2.2 i K 4.2.3.

Korak 4.2.3. se ponavlja se dok ne nastupi situacija iz koraka 4.2.1 ili 4.2.2 kada se posupak identifikacije najkraćeg puta između čvorova i i j završava a brojevi upisani u tabelu predstavljaju oznake (idekse) čvorova preko kojih ide najkraći put p_{ij} sa dužinom $d(p_{ij}) = c_{ij}^n$. Ovaj postupak identifikacije najkraćih puteva može se izvesti za svako $i \in N$ i svako $j \in N$ i na taj način mogu se identifikovati najkraći putevi između svaka dva čvora i i j date mreže.

PRIMER 71:

Za datu mrežu naći dužine najkraćih puteva između svaka dva čvora



Rešenje:

$$\begin{aligned}
 k &= 0 \\
 c_{ij}^0 &= c_{ij} \quad \forall (i, j) \in L \\
 c_{ii}^0 &= 0 \quad \forall i \in N \\
 c_{ij}^0 &= \infty \quad \forall (i, j) \notin L
 \end{aligned}$$

$k = 3, I = \{1\}, J = \{4, 5\}$, posmatramo:

$$c_{14}^3 = \min \{c_{14}^2, c_{13}^2 + c_{34}^2\} = \min \{8, 4-3\} = 1$$

$$c_{15}^3 = \min \{c_{15}^2, c_{13}^2 + c_{35}^2\} = \min \{\infty, 4+2\} = 6$$

$$\begin{array}{c}
 I \\
 \downarrow \\
 C^\circ = C = \begin{array}{c} J \rightarrow \\
 \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{array} \begin{array}{c} \left[\begin{array}{cccccccc}
 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
 0 & 2 & 4 & \infty & \infty & \infty & \infty & \infty \\
 \infty & 0 & \infty & 6 & \infty & \infty & \infty & \infty \\
 \infty & \infty & 0 & -3 & 2 & \infty & \infty & \infty \\
 \infty & \infty & \infty & 0 & 3 & 3 & 7 & \infty \\
 \infty & \infty & \infty & -1 & 0 & \infty & 8 & \infty \\
 \infty & \infty & \infty & \infty & \infty & 0 & 5 & 2 \\
 \infty & \infty & \infty & \infty & \infty & -3 & 0 & -5 \\
 \infty & \infty & \infty & \infty & \infty & 4 & \infty & 0 \end{array} \right]
 \end{array}
 \end{array}$$

$$k = 1$$

$I = \emptyset$ (elementi k -te kolone različiti od ∞ , a da se ne nalaze na dijagonali),

$J = \{2, 3\}$ (elementi k -tog reda različiti od ∞ , a da se ne nalaze na dijagonali)

Pošto je $I = \emptyset$, matrica C^1 će ostati ista kao C° .

$k = 2, I = \{1\}, J = \{4\}$, razmatramo samo element:

$$c_{14}^2 = \min \{c_{14}^1, c_{12}^1 + c_{24}^1\} = \min \{\infty, 2+6\} = 8$$

C^2 se razlikuje od C° i C^1 samo u elementu c_{14} koji je bio ∞ , a sada je 8.

$$c_{36}^4 = \min \{c_{36}^3, c_{34}^3 + c_{46}^3\} = \min \{\infty, -3+3\} = 0$$

$$c_{37}^4 = \min \{c_{37}^3, c_{34}^3 + c_{47}^3\} = \min \{\infty, -3+7\} = 4$$

$$c_{55}^4 = \min \{c_{55}^3, c_{54}^3 + c_{45}^3\} = \min \{0, -1+3\} = 0$$

$$c_{56}^4 = \min \{c_{56}^3, c_{54}^3 + c_{46}^3\} = \min \{\infty, -1+3\} = 2$$

$$c_{57}^4 = \min \{c_{57}^3, c_{54}^3 + c_{47}^3\} = \min \{8, -1+7\} = 6$$

$$C^4 = \begin{bmatrix} 0 & 2 & 4 & 1 & 4 & 4 & 8 & \infty \\ \infty & 0 & \infty & 6 & 9 & 9 & 13 & \infty \\ \infty & \infty & 0 & -3 & 0 & 0 & 4 & \infty \\ \infty & \infty & \infty & 0 & 3 & 3 & 7 & \infty \\ \infty & \infty & \infty & -1 & 0 & 2 & 6 & \infty \\ \infty & \infty & \infty & \infty & \infty & 0 & 5 & 2 \\ \infty & \infty & \infty & \infty & \infty & -3 & 0 & -5 \\ \infty & \infty & \infty & \infty & \infty & 4 & \infty & 0 \end{bmatrix}$$

Zbog opširnosti preskačemo nekoliko iteracija. Za $k = 7$ se dobija sledeća matrica rastojanja:

$$C^7 = \begin{bmatrix} 0 & 2 & 4 & 1 & 4 & 4 & 8 & 3 \\ \infty & 0 & \infty & 6 & 9 & 9 & 13 & 8 \\ \infty & \infty & 0 & -3 & 0 & 0 & 4 & -1 \\ \infty & \infty & \infty & 0 & 3 & 3 & 7 & 2 \\ \infty & \infty & \infty & -1 & 0 & 2 & 6 & 1 \\ \infty & \infty & \infty & \infty & \infty & 0 & 5 & 0 \\ \infty & \infty & \infty & \infty & \infty & -3 & 0 & -5 \\ \infty & \infty & \infty & \infty & \infty & 4 & 9 & 0 \end{bmatrix}$$

Dalje, imamo za $k = 8$ $I = \{1,2,3,4,5,6,7\}, J = \{6,7\}$. Za sledeće elemente matrice C^8 izračunavamo nove vrednosti:

$$C_{16}^8 = 4, C_{17}^8 = 8,$$

$$C_{26}^8 = 9, C_{27}^8 = 13,$$

$$C_{36}^8 = 0, C_{37}^8 = 4,$$

$$C_{46}^8 = 3, C_{47}^8 = 7$$

$$, C_{56}^8 = 2, C_{57}^8 = 6,$$

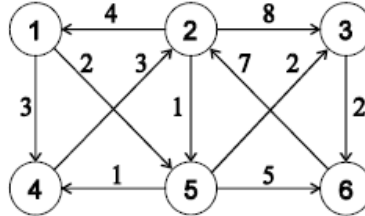
$$C_{66}^8 = 0, C_{67}^8 = -5,$$

$$C_{76}^8 = -3, C_{77}^8 = 0.$$

Primitimo da je matrica $C^8 = C^7$. Pošto je $k = n$ i svi $c_{ii}^8 = 0, i = 1, \dots, 8$ sledi da je algoritam došao do kraja. Poslednja matrica predstavlja rešenje zadatka. Ostalo je da rekonstruišemo putanju (na času formirati matricu D i rekonstrukciju najkraćeg puta).

PRIMER 72

Data je mreža. Naći najkraće puteve između svaka dva čvora u mreži



Rešenje:

$$C^0 = C = \begin{bmatrix} 0 & \infty & \infty & 3 & 2 & \infty \\ 4 & 0 & 8 & \infty & 1 & \infty \\ \infty & \infty & 0 & \infty & \infty & 2 \\ \infty & 3 & \infty & 0 & \infty & \infty \\ \infty & \infty & 2 & 1 & 0 & 5 \\ \infty & 7 & \infty & \infty & \infty & 0 \end{bmatrix}$$

$$k = 1, I = \{2\}, J = \{4, 5\}$$

$$c_{24} = \min\{\infty, 4+3\} = 7^*$$

$$c_{25} = \min\{1, 4+2\} = 1$$

U matrici je promenjena vrednost jedino elementa c_{24} označen * i to sa ∞ na 7. Svi ostali elementi matrice C^1 su isti kao i elementi matrice C^0 .

$$k = 2, I = \{4, 6\}, J = \{1, 3, 4, 5\}$$

$$c_{41} = \min\{\infty, 3+4\} = 7^*$$

$$c_{43} = \min\{\infty, 3+8\} = 11^*$$

$$c_{44} = \min\{0, 3+7\} = 0$$

$$c_{45} = \min\{\infty, 3+1\} = 4^*$$

$$c_{61} = \min\{\infty, 7+4\} = 11^*$$

$$c_{63} = \min\{\infty, 7+8\} = 15^*$$

$$c_{64} = \min\{\infty, 7+7\} = 14^*$$

$$c_{65} = \min\{\infty, 7+1\} = 8^*$$

Matrica C^2 ima oblik:

$$C^2 = \begin{bmatrix} 0 & \infty & \infty & 3 & 2 & \infty \\ 4 & 0 & 8 & 7 & 1 & \infty \\ \infty & \infty & 0 & \infty & \infty & 2 \\ 7 & 3 & 11 & 0 & 4 & \infty \\ \infty & \infty & 2 & 1 & 0 & 5 \\ 11 & 7 & 15 & 14 & 8 & 0 \end{bmatrix}$$

U narednim koracima biće date samo matrice:

$$C^3 = \begin{bmatrix} 0 & \infty & \infty & 3 & 2 & \infty \\ 4 & 0 & 8 & 7 & 1 & 10 \\ \infty & \infty & 0 & \infty & \infty & 2 \\ 7 & 3 & 11 & 0 & 4 & 13 \\ \infty & \infty & 2 & 1 & 0 & 4 \\ 11 & 7 & 15 & 14 & 8 & 0 \end{bmatrix}$$

$$C^4 = \begin{bmatrix} 0 & 6 & 14 & 3 & 2 & 16 \\ 4 & 0 & 8 & 7 & 1 & 10 \\ \infty & \infty & 0 & \infty & \infty & 2 \\ 7 & 3 & 11 & 0 & 4 & 13 \\ 8 & 4 & 2 & 1 & 0 & 4 \\ 11 & 7 & 15 & 14 & 8 & 0 \end{bmatrix}$$

$$C^5 = \begin{bmatrix} 0 & 6 & 4 & 3 & 2 & 6 \\ 4 & 0 & 3 & 2 & 1 & 5 \\ \infty & \infty & 0 & \infty & \infty & 2 \\ 7 & 3 & 6 & 0 & 4 & 8 \\ 8 & 4 & 2 & 1 & 0 & 4 \\ 11 & 7 & 10 & 9 & 8 & 0 \end{bmatrix}$$

$$C^6 = \begin{bmatrix} 0 & 6 & 4 & 3 & 2 & 6 \\ 4 & 0 & 3 & 2 & 1 & 5 \\ 13 & 9 & 0 & 11 & 10 & 2 \\ 7 & 3 & 6 & 0 & 4 & 8 \\ 8 & 4 & 2 & 1 & 0 & 4 \\ 11 & 7 & 10 & 9 & 8 & 0 \end{bmatrix}$$

Poslednja matrica je rešenje zadatka.

Rekonstrukcija rešenja.