Aplicatii rezolvate - Distante si drumuri minime –

APLICATIE: Pentru graful orientat ponderat (G, c) reprezentat prin matricea costurilor:

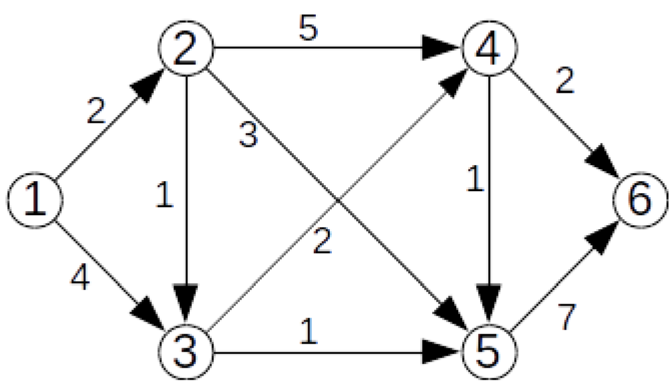
$$A = \begin{pmatrix} 0 & 2 & 4 & \infty & \infty & \infty \\ \infty & 0 & 1 & 5 & 3 & \infty \\ \infty & \infty & 0 & 2 & 1 & \infty \\ \infty & \infty & \infty & 0 & 1 & 2 \\ \infty & \infty & \infty & \infty & 0 & 7 \\ \infty & \infty & \infty & \infty & \infty & 0 \end{pmatrix}$$

luand ca nod sursa nodul s = 1, aplicati *Algoritmului Dijkstra*.

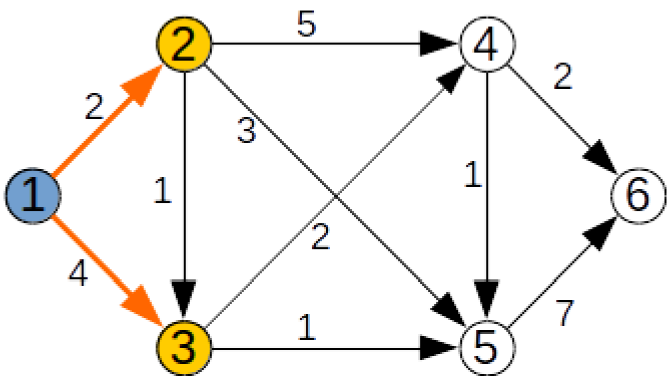
SOLUTIE: Reprezentam graful. Folosim următoarele structuri de date:

- un vector t[], în care t[k] reprezintă costul minim curent al drumului de la nodul sursă s=1 la k;
- un vector caracteristic S[], în care S[k]=1 dacă pentru nodul k s-a determinat costul minim final, respectiv S[k]=0 dacă pentru nodul k nu s-a determinat (încă) acest cost;
- Pentru determinarea drumurilor minime de la nodul s la nodurile grafului vom utiliza si un vector T AT A[] avand semnificatia T AT A[k] = nodul j ce este predecesorul direct al nodului k pe drumul minim de la s la k, ∀ k ∈ {1, . . . , n}.

Graful dat este:

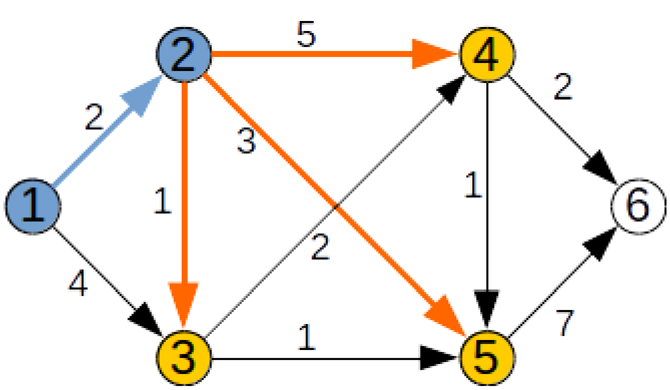


Pasul 0: Initializăm vectorii, ca mai jos. Inițial în mulțimea S se află doar nodul sursă s=1.



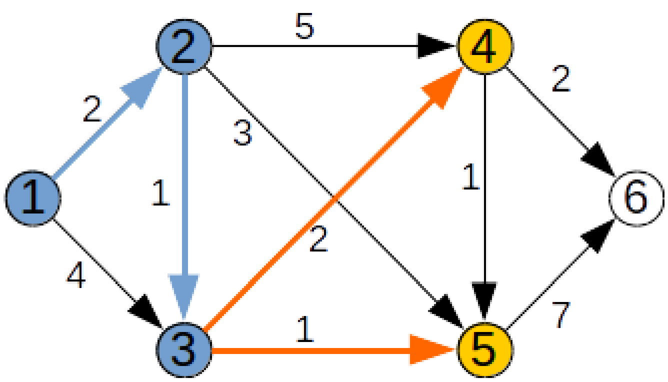
k	1	2	3	4	5	6
S[k]	1	0	0	0	0	0
t[k]	0	2	4	∞	∞	∞
TATA[k]	0	∞	∞	∞	∞	∞

Pasul 1: Alegem un vârf k din afara lui S, pentru care t[k] este finit și minim. Acesta este k=2. Îl adăugăm în S și analizăm nodurile x pentru care (k,x) este arc. Se vor relaxa nodurile 3 4 5, adica pentru succesorii nodului k reactualizam vectorul t.



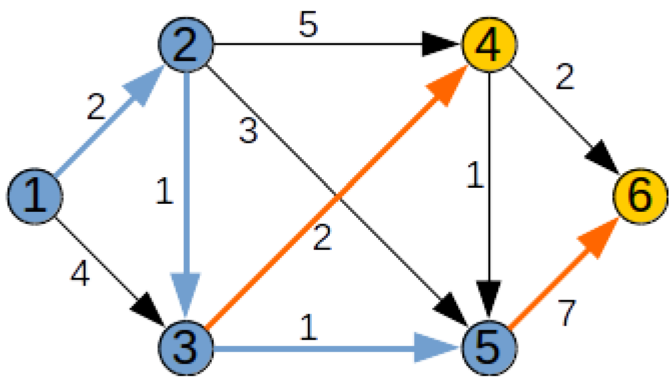
k	1	2	3	4	5	6
S[k]	1	1	0	0	0	0
t[k]	0	2	3	7	5	∞
TATA[k]	0	1	∞	∞	∞	∞

Pasul 2: Alegem un vârf k din afara lui S, pentru care d[k] este finit și minim. Acesta este k=3. Îl adăugăm în S și analizăm nodurile x pentru care (k,x) este arc. Se vor relaxa nodurile 4 5.



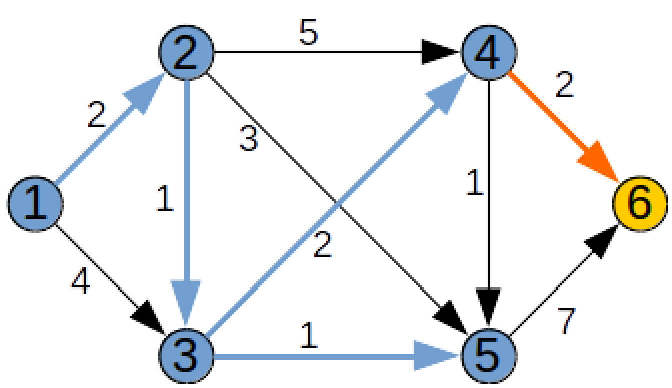
k	1	2	3	4	5	6
S[k]	1	1	0	0	0	0
t[k]	0	2	3	5	4	∞
TATA[k]	0	1	2	∞	∞	∞

Pasul 3: Alegem un vârf k din afara lui S, pentru care t[k] este finit și minim. Acesta este k=5. Îl adăugăm în S și analizăm nodurile x pentru care (k,x) este arc. Se va relaxa nodul 6.



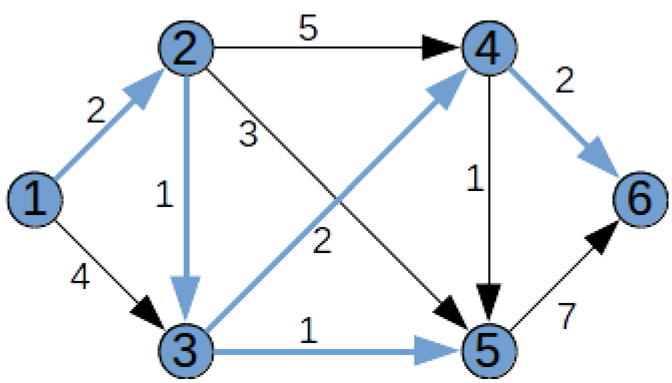
k	1	2	3	4	5	6
S[k]	1	1	0	0	0	0
t[k]	0	2	3	5	4	11
TATA[k]	0	1	2	∞	3	∞

Pasul 4: Alegem un vârf k din afara lui S, pentru care t[k] este finit și minim. Acesta este k=4. Îl adăugăm în S și analizăm nodurile x pentru care (k,x) este arc. Se va relaxa nodul 6.



k	1	2	3	4	5	6
S[k]	1	1	0	0	0	0
t[k]	0	2	3	5	4	7
TATA[k]	0	1	2	3	3	∞

Pasul 5: Alegem un vârf k din afara lui S, pentru care t[k] este finit și minim. Acesta este k=6. Îl adăugăm în S și analizăm nodurile x pentru care (k,x) este arc. Nu mai există asemenea arce, niciun nod nu se mai relaxează.



k	1	2	3	4	5	6
S[k]	1	1	0	0	0	0
t[k]	0	2	3	5	4	7
TATA[k]	0	1	2	3	3	4

Algoritmul lui Dijkstra s-a încheiat. Valorile finale din vectorul t[] – distanțele minime de la nodul s=1 la toate celelalte sunt cele de mai sus.

Drumurile minime se gasesc pentru fiecare nod mergand inapoi de-a lungul vectorului TATA[ ] pana ajungem in nodul sursa.

De exemplu pt nodul x=6 : 6-4-3-2-1, deci drumul minim determinat de algoritm de la 1 la 6 este [1,2,3,4,6]. Analog, drumurile minime determinate de algoritm sunt:

- de la 1 la 1 : [1]
- de la 1 la 2: [1,2]
- de la 1 la 3: [1,2,3]
- de la 1 la 4: [1,2,3,4]
- de la 1 la 5: [1,2,3,5]