

## 2 PROIECTELE LECȚIILOR

Prof. Adriana Chereș

Stud. Jaclina-Iana Bulat, 931

**Data:** 22.03.2024

**Clasa:** a XI-a

**Disciplina:** Informatică

**Unitatea de învățare:** Grafuri ponderate

**Lecția:** Algoritmul lui Dijkstra

**Tipul lecției:** Transmitere și asimilare de noi cunoștințe

### Competențe specifice:

1.1. Transpunerea unei probleme din limbaj natural în limbaj de grafuri, folosind corect terminologia specifică

1.4. Descrierea algoritmilor fundamentali de prelucrare a grafurilor și implementarea acestora într-un limbaj de programare

1.8. Aplicarea în mod creativ a algoritmilor fundamentali în rezolvarea unor probleme concrete

**Obiective operaționale:** La sfârșitul lecției, elevul va fi capabil să:

O1. Înțelege conceptul algoritmului lui Dijkstra

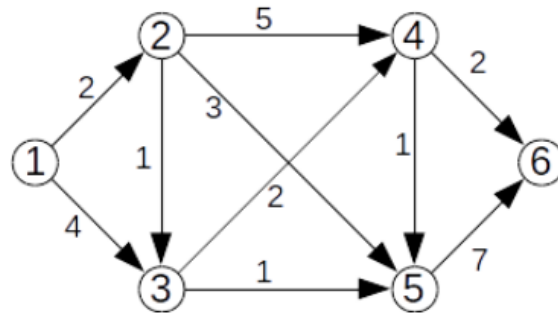
O2. Interpreteze corect un graf ponderat și să determine corect dacă e sau nu cazul pentru algoritmul lui Dijkstra

O3. Să implementeze corect algoritmul

Evenimentele lecției	Activitatea din lecție	Strategia didactică și evaluarea
Captarea atenției	Profesorul începe lecția prin organizarea clasei, și situații cotidiene care ar necesita astfel de rezolvări. (călătoritul cu avionul, mașina, etc)	Conversația
Enunțarea obiectivelor	Profesorul explică clar obiectivele lecției: Înțelegerea conceptului de cost minim pe un graf ponderat, de ce folosim algoritmul lui Dijkstra. Deducerea din cerință a faptului ca trebuie folosit un algoritm de cost minim.	Descrierea
Prezentarea de material nou/ sarcinilor de învățare	Profesorul introduce conceptul de algoritm de rezolvare a unei probleme care cere drumul de cost minim și asocierea sa cu Greedy.  Considerăm un graf $G$ cu $n$ perechi de arce și un nod inițial $s$ .  În cadrul acestui algoritm, ne propunem să alegem nodul pentru care drumul de la $s$ la $x$ (nodul țintă) va fi de cost minim.  Pașii sunt:	Descrierea Exemplificarea

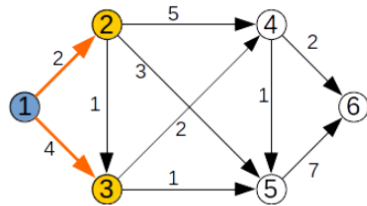
Inițializăm cei doi vectori  $d$  (costul minim curent al drumului de la nodul sursă la  $k$ ) și  $F$  (vectorul caracteristic pentru care avem  $F[k] = 1$  dacă s-a determinat costul minim final sau 0 dacă nu s-a determinat încă acest cost)

Luăm un exemplu concret:



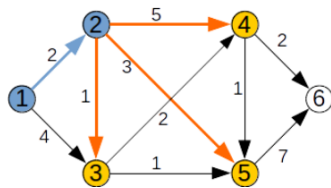
Discutăm fiecare iterație în parte:

**Pasul 0:** Inițializăm cei doi vectori, ca mai jos. Inițial în mulțimea  $F$  se află doar nodul sursă  $s=1$ .



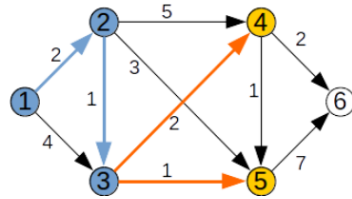
k	1	2	3	4	5	6
$v[k]$	1	0	0	0	0	0
$d[k]$	0	2	4	$\infty$	$\infty$	$\infty$

**Pasul 1:** Alegem un vârf  $k$  din afara lui  $F$ , pentru care  $d[k]$  este finit și minim. Acesta este  $k=2$ . Îl adăugăm în  $F$  și analizăm nodurile  $x$  pentru care  $(k,x)$  este arc. Se vor relaxa nodurile 3 4 5.



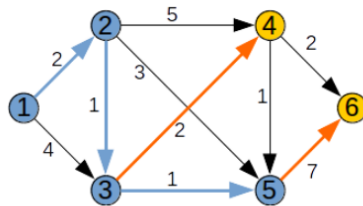
k	1	2	3	4	5	6
$v[k]$	1	1	0	0	0	0
$d[k]$	0	2	3	7	5	$\infty$

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



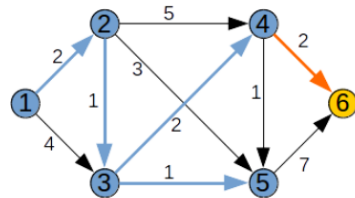
k	1	2	3	4	5	6
$v[k]$	1	1	0	0	0	0
$d[k]$	0	2	3	5	4	$\infty$

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



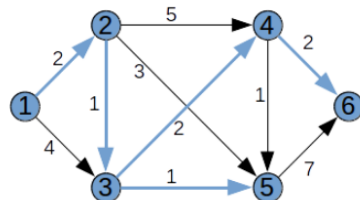
k	1	2	3	4	5	6
$v[k]$	1	1	0	0	0	0
$d[k]$	0	2	3	5	4	11

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



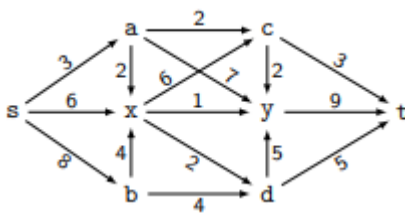
k	1	2	3	4	5	6
$v[k]$	1	1	0	0	0	0
$d[k]$	0	2	3	5	4	7

**Pasul 5:** Alegem un vârf  $k$  din afara lui  $F$ , pentru care  $d[k]$  este finit și minim. Acesta este  $k=6$ . Îl adăugăm în  $F$  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
$v[k]$	1	1	0	0	0	0
$d[k]$	0	2	3	5	4	7

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

	<pre> #define INFINIT 1000000000 ... //nodul sursa este s ... for(i =1 ; i &lt;= n ; i ++ ) {     f[i] = 0;     d[i] = a[s][i]; }  f[s] = 1, d[s] = 0; d[0] = INFINIT; // pentru determinarea nodului cu costul minim for(int k = 1 ; k &lt; n ; ++k) {     int pmax = 0;     for(i = 1 ; i &lt;= n ; ++i)         if(f[i] == 0 &amp;&amp; d[i] &lt; d[pmax])             pmax = i;      if(pmax &gt; -1)     {         f[pmax] = 1;         for(i = 1; i &lt;= n ; ++i)             if(f[i] == 0 &amp;&amp; d[i] &gt; d[pmax] + a[pmax][i])                 d[i] = d[pmax] + a[pmax][i];     } } </pre>	
Dirijarea învățării	<p>Elevii sunt invitați să rezolve ei singuri pe un graf nou cu aceeași cerință.</p> <ol style="list-style-type: none"> <li>Se dă un graf cu n perechi de muchii și costuri asociate, și un s sursa de unde pornim. Să se stabilească drumul de cost minim de la s la celelalte noduri și să se afișeze cel mai lung drum. (Suntem pe munte și dorim să parcurgem cel mai lung și interesant traseu pentru a ne bucura de drumeție)</li> </ol> 	Conversația Exercițiul Analizarea
Asigurarea reținerii	<p>Profesorul recapitulează principalele concepte prezentate pe parcursul lecției. Întreabă elevii dacă există neclarități.</p> <p>Temă:</p> <p>Se da un graf ponderat. Se citesc n - nr de varfuri, m- nr de arce , arcele si costurile acestora.</p> <ol style="list-style-type: none"> <li>Afisati perechile de varfuri intre care exista legaturi directe si care au lungime minima.</li> <li>Care este costul total si costul mediu al grafului.</li> <li>Se citesc 3 noduri, X,Y si Z. Sa se verifice care dintre nodurile X si Y este mai apropiat de nodul Z. Daca nu se poate ajunge din ele in Z, se va afisa un mesaj.</li> <li>Se citeste un numar t. Sa se afiseze toate perechile de noduri</li> </ol>	Conversația

	(i,j) pentru care drumul minim din i in j e mai mic decat t e) Sa se afiseze perechile de noduri pentru care nu exista drum decat intr-un sens. f) Sa se afiseze pentru fiecare nod, nodurile in care nu se poate ajunge plecand din el. g) Sa se afiseze pentru fiecare nod i, cele mai apropiate 2 noduri in care se poate ajunge plecand din acesta. h) Se citesc doua noduri x si y. Sa se afiseze, daca exista, drumul de cost minim de la x la y.	
--	---	--

**Data: 22.03.2024**

**Clasa:** a XI-a

**Disciplina:** Informatică

**Unitatea de învățare:** Grafuri ponderate

**Lección:** Algoritmul Roy-Warshall -Floyd

**Tipul lecției:** Transmitere și asimilare de noi cunoștințe

**Competențe specifice:**

1.1. Transpunerea unei probleme din limbaj natural în limbaj de grafuri, folosind corect terminologia specifică

1.4. Descrierea algoritmilor fundamentali de prelucrare a grafurilor și implementarea acestora într-un limbaj de programare

1.8. Aplicarea în mod creativ a algoritmilor fundamentali în rezolvarea unor probleme concrete

**Obiective operaționale:** La sfârșitul lecției, elevul va fi capabil să:

O1. Înțelege conceptul algoritmilor

O2. Interpreteze corect un graf ponderat și să determine corect dacă e sau nu cazul pentru algoritmul studiat

O3. Să implementeze corect algoritmi

Evenimentele lecției	Activitatea din lecție	Strategia didactică și evaluarea
Captarea atenției	Profesorul începe lecția prin organizarea clasei, și situații cotidiene care ar necesita astfel de rezolvări. (călătoritul cu avionul, mașina, etc)	Conversația
Enunțarea obiectivelor	Profesorul explică clar obiectivele lecției: Înțelegerea conceptului de cost minim pe un graf ponderat, de ce folosim algoritmi. Deducerea din cerință a faptului ca trebuie folosit un algoritm de cost minim. Înțelegerea conceptului de matricea drumurilor.	Descrierea
Prezentarea de material nou/ sarcinilor de învățare	Profesorul introduce conceptul de algoritm de rezolvare a unei probleme care cere matricea drumurilor respectiv drumul cu cost minim.	Descrierea Exemplificarea

Algoritmul se regăsește sub diferite denumiri care conțin numele descoperitorilor, este bazat pe programarea dinamică și poate fi utilizat în următoarele două moduri:

- pentru un graf orientat oarecare determină **matricea drumurilor** – stabilește despre oricare două noduri  $x$   $y$  dacă există drum de la  $x$  la  $y$  – este de regulă cunoscut sub numele **Roy-Warshall**
- pentru un graf orientat ponderat (cu costuri) – determină pentru fiecare pereche de noduri costul minim al unui drum cu extremitățile în acele noduri – este de regulă cunoscut sub numele **Roy-Floyd**

Considerăm un graf  $G$  cu  $n$  perechi de arce și costurile aferente.

<https://www.pbinfo.ro/probleme/589/roy-floyd>

Se dorește determinarea pentru fiecare pereche de noduri  $x$   $y$ , dacă există, a unui drum de cost minim – în care suma costurilor asociate arcelor care definesc drumul este minimă.

Algoritmul pornește **matricea costurilor**,  $A$  – în care:

$$A_{i,j} = \begin{cases} 0 & \text{dacă } i = j, \\ \text{costul arcului } (i, j) & \text{dacă există arc de la } i \text{ la } j, \\ \infty & \text{dacă nu există arc de la } i \text{ la } j \end{cases}$$

Prin algoritmul Roy-Floyd matricea va fi transformată, astfel încât la final va avea următoarea semnificație:

$$D_{i,j} = \begin{cases} 0 & \text{dacă } i = j, \\ \text{costul minim al unui drum de la } i \text{ la } j & \text{dacă există un asemenea drum,} \\ \infty & \text{dacă nu există drum de la } i \text{ la } j \end{cases}$$

Pașii sunt:

Inițializăm matricea costurilor conform condițiilor de mai sus

Transformăm matricea cu ajutorul algoritmului Roy-Floyd pentru a stabili costurile minime.

```
//D[][] este inițial matricea costurilor arcelor
for(int k = 1 ; k <= n ; k ++ )
    for(int i = 1 ; i <= n ; i ++ )
        for(int j = 1 ; j <= n ; j ++ )
            if(D[i][j] > D[i][k] + D[k][j])
                D[i][j] = D[i][k] + D[k][j];
```

Dirijarea învățării	<p>Elevii sunt invitați să rezolve ei singuri o nouă problemă.</p> <p>Se da un graf orientat tare conex cu <math>n</math> varfuri și <math>m</math> arce prin lista arcelor. Se numește varf central un varf cu proprietatea că suma distanțelor de la el la toate celelalte varfuri este minim. Distanța de la <math>i</math> la <math>j</math> este egală cu lungimea celui mai scurt drum de la <math>i</math> la <math>j</math>. Afișați varfurile centrale ale grafului. Se va folosi algoritmul Roy-Floyd pentru arce de cost 1.</p> <p>Exemplu:</p> <pre> graf.in 8 13 1 2 1 3 2 3 3 4 4 5 5 6 6 7 8 7 7 8 8 1 6 3 6 2 6 4 graf.out 6 </pre>	Conversația Exercițiul Analizarea
Asigurarea reținerii	<p>Profesorul recapitulează principalele concepte prezentate pe parcursul lecției. Întreabă elevii dacă există neclarități.</p> <p>Temă:</p> <p>Se da un graf orientat cu <math>n</math> varfuri și <math>m</math> arce având arcele etichetate cu costuri numere naturale. Se citesc apoi două varfuri <math>x</math> și <math>y</math>. Afișați drumul de cost minim de la varful <math>x</math> la varful <math>y</math> trecând prin varful <math>y</math>, precum și costul acestui drum.</p> <p>Exemplu:</p> <pre> date.in 12 21 (n,m) 1 2 20 (arcele și costurile) 1 3 35 1 7 20 2 4 30 3 4 40 3 6 40 3 8 80 4 5 25 5 6 5 6 8 30 6 9 10 </pre>	Conversația

	7 8 15 7 11 100 8 9 40 8 10 30 8 11 35 9 10 30 10 12 25 11 12 10 10 5 15 5 1 10 1 10 (x,y) date.out 1 7 8 10 5 1 90	
--	--	--