Grafuri ponderate. Aplicaţii

Analiza algoritmilor

-laborator-

conf. dr. ing. Ciprian-Bogdan Chirilă

Cuprins

- Definiţii
- Arbore de acoperire minimi
  - Algoritmul lui Prim
  - Cautare bazată pe prioritate
  - Algoritmul lui Kruskal
- Determinarea drumurilor minime
- Aplicaţie: modernizarea unei reţele de transport.

Ce este un graf ponderat ?

- Un graf ponderat “weighted graph” este un graf în cadrul căruia fiecarui arc îi este asociată o valoare;
- Valoarea asociată arcului are semnificaţia de “cost” a legăturii între cele 2 noduri sau de “distanţa” între noduri.

Arbore de acoperire minim

Fie G=(N,A) un graf conex în care fiecare arc are ataşat un cost c(u,v).

Un arbore de acoperire a lui G este arborele liber care:

- cuprinde toate nodurile din N;
- este subgraf a lui G.

Algoritmul lui Prim (principiu)

- Fie G=(N,A) şi o funcţie de cost definită pe arcele sale;
- Fie U o submulţime a lui N;
- Dacă (u,v) este un arc cu costul cel mai scăzut, a.î. u € U şi v € N\U atunci există un arbore de acoperire minim care include (u,v).

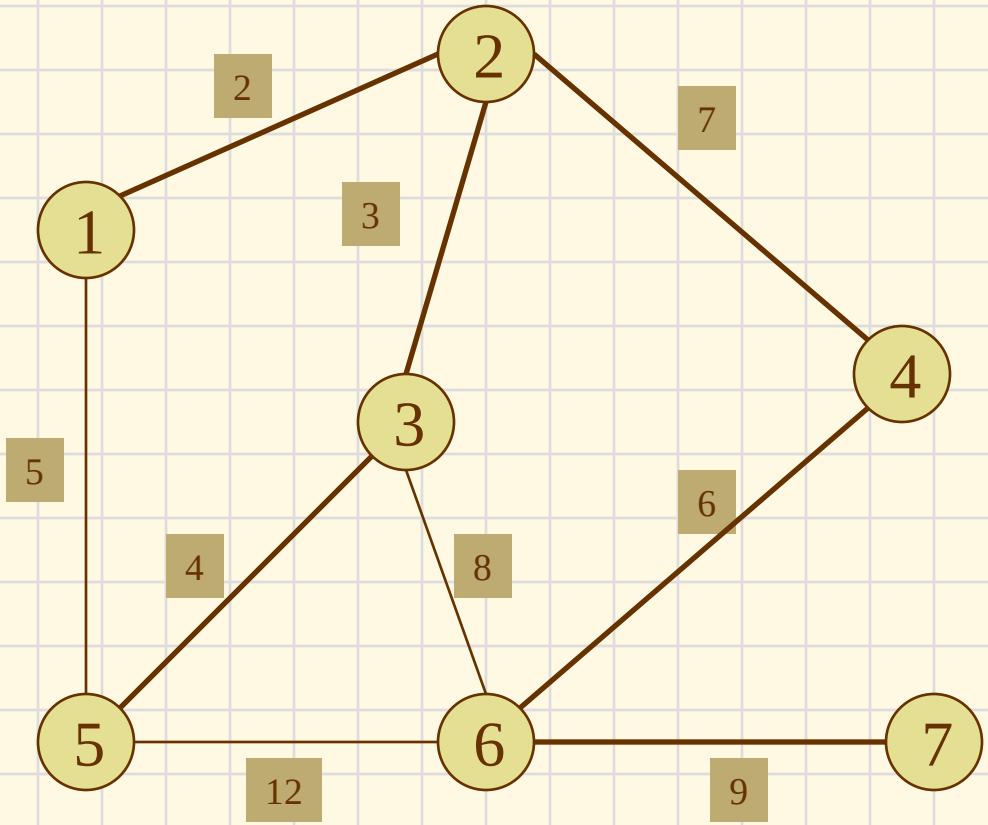
Algoritmul lui Prim (limbaj natural)

1. Introducem un nod arbitrar în U.
2. Selectăm arcul de cost minim (u,v) care conectează mulţimea U cu N\U, u<U,v<N\U. Adăugăm acest arc arborelui.
3. Dacă U!=N executăm pasul 2.

Algoritmul lui Prim (pseudocod)

```
procedure PRIM(G:graf; var T:MultimeArce);
{Date de intrare: graful ponderat G=(N,A)}
{Date de ieşire: arborele de acoperire minim T al grafului G, T:MultimeArce}
begin
 *iniţializează mulţimea arcelor arborelui T ca fiind mulţimea vidă
 *iniţializează mulţimea nodurilor arborelui, U, prin adăugarea unui nod arbitrar de pornire
 while (*nu s-au adăugat mulţimii U toate nodurile mulţimii N)
 begin
  *găseşte arcul (u,v) cu costul cel mai redus care conectează pe U cu N\U şi care are
     capetele u în U şi pe v în N\U
  *adaugă arcul (u,v) la arborele de acoperire T
  *adaugă nodul v la mulţimea U a nodurilor arborelui T
 end;
end;
```

Algoritmul lui Prim (demo)



U={1}; N\U={2,3,4,5,6,7}

U={1,2}; N\U={3,4,5,6,7}

U={1,2,3}; N\U={4,5,6,7}

U={1,2,3,5}; N\U={4,6,7}

U={1,2,3,4,5}; N\U={6,7}

U={1,2,3,4,5,6}; N\U={7}

U={1,2,3,4,5,6,7}; N\U=ø

Căutare bazată pe prioritate

- Nodurile se împart în trei clase:
- Clasa arbore – noduri vizitate;
- Clasa vecinătate – noduri adiacente celor din clasa arbore;
- Clasa neîntalnite – noduri la care nu s-a ajuns încă;

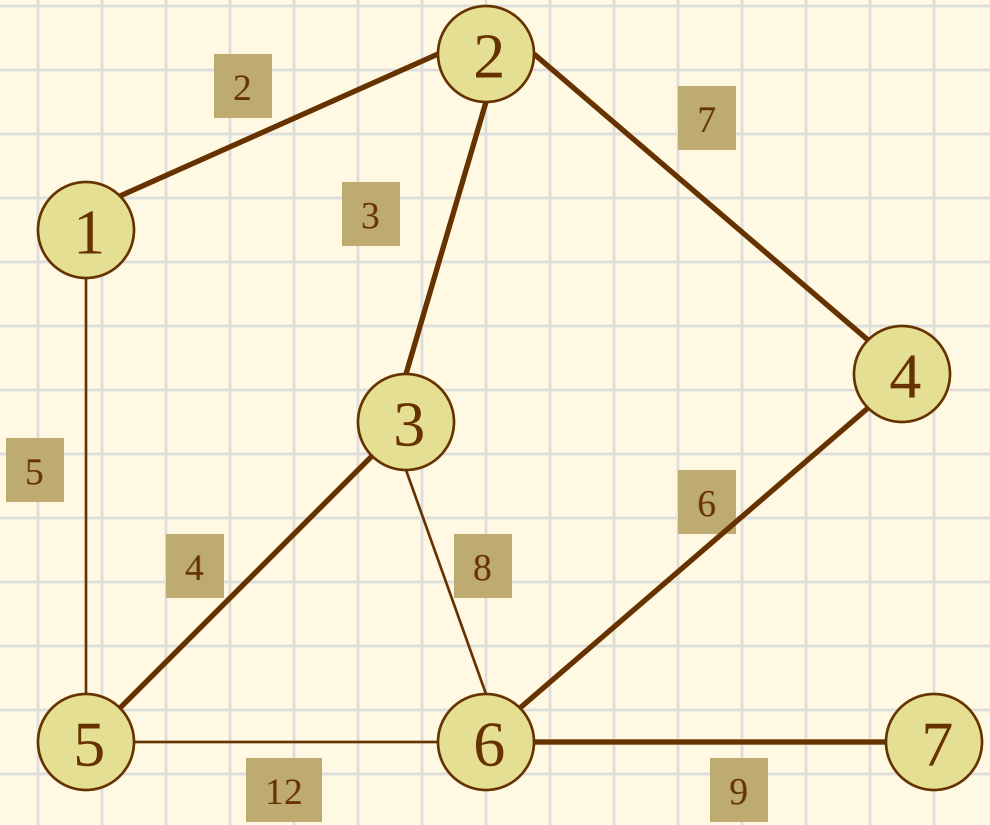
Algoritmul lui Kruskal (principiu)

- Fie graful G=(N,A).
- La fiecare pas se alege arcul de cost minim;
- Dacă arcul ales nu introduce ciclu în arborele de acoperire minim atunci el se adaugă la acesta.

Algoritmul lui Kruskal (algoritm)

```
procedure KRUSKAL(G:graf; var T:MultimeArce);
{Date de intrare: graful ponderat G=(N,A)}
{Date de ieşire: arborele de acoperire minim T al grafului G, T:MultimeArce}
begin
 *iniţializează mulţimea arcelor arborelui T ca fiind mulţimea vidă
 *iniţializează mulţimea de componente conexe a.î. fiecare componentă conexă
     este alcătuită dintr-un nod al lui G
 *sortează arcele lui G în ordine crescătoare a costului înserându-le într-o coadă cu
     priorităţi
 while(*există mai multe componente conexe) do
 begin
  * scoate arcul de cost minim din coadă, fie acesta (x,y)
  * if (* x şi y aparţin la componente conexe diferite) then
   *adaugă arc(x,y) la T şi uneşte cele două componente
 end;
end;
```

Algoritmul lui Kruskal (demo)



Adaug arcul (1,2) cost 2

Adaug arcul (2,3) cost 3

Adaug arcul (3,5) cost 4

Nu adaug arcul (1,5)!!!

Adaug arcul (4,6) cost 6

Adaug arcul (2,4) cost 7

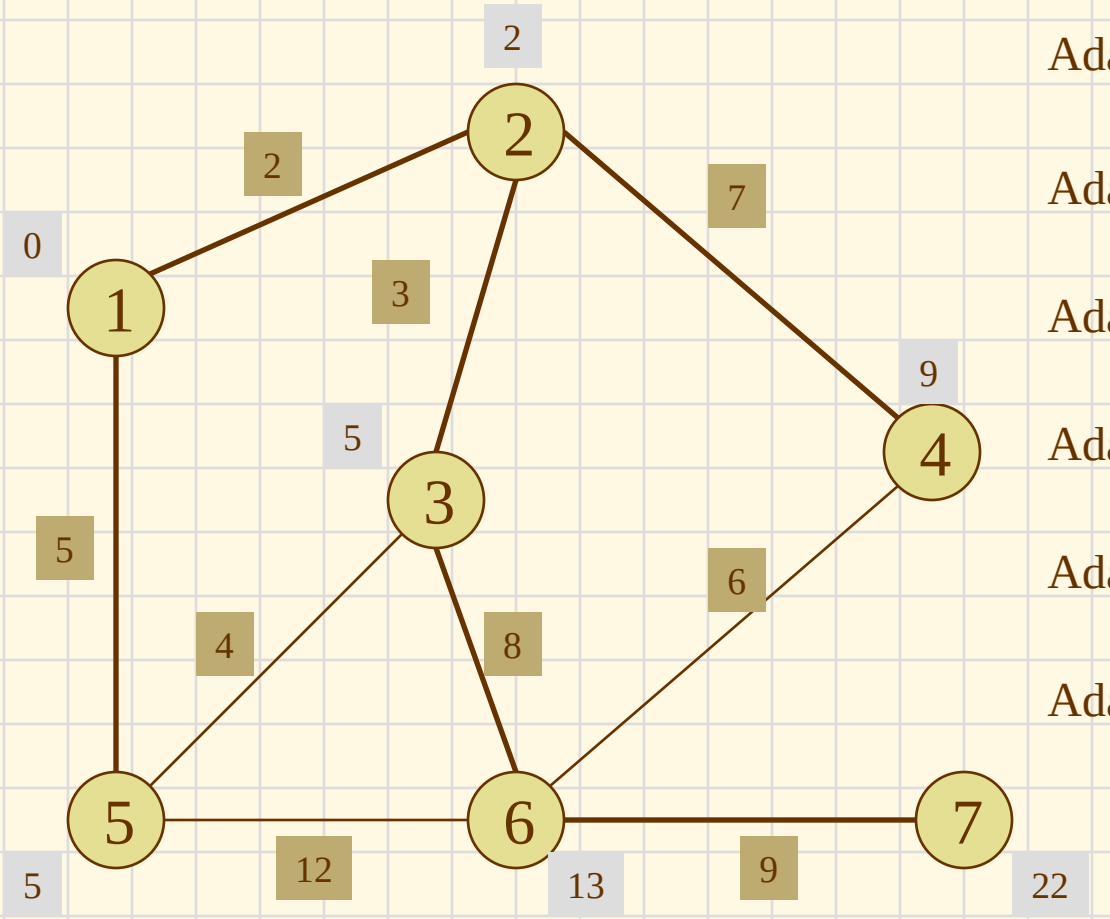
Adaug arcul (6,7) cost 9

Nu adaug arcele (3,6); (5,6)!!!

Determinarea drumurilor minime

- Pentru determinarea arborelui de acoperire minim s-a selectat nodul:
  - cel mai apropiat de arbore
- În cazul arborelui corespunzător drumurilor minime cu originea în x se va selecta nodul:
  - cel mai aproape de nodul x

Determinarea drumurilor minime (demo)



Adaug arcul (1,2) cost 2

Adaug arcul (1,5) cost 5

Adaug arcul (2,3) cost 3

Adaug arcul (2,4) cost 7

Adaug arcul (3,6) cost 8

Adaug arcul (6,7) cost 9

Aplicaţie

O companie construieşte o reţea de şosele.

Se propun 3 soluţii care urmăresc următoarele aspecte:

- minimizarea costului lucrărilor;
- minimizare costului de acces din capitală la oricare din localităţile judeţului;
- o variantă de compromis;

Concluzii

- 3 metode de determinare al arborelui de acoperire minim;
- 1 metodă de determinare a drumurilor minime;
- 1 metodă de compromis – arborele de acoperire minim cu rădăcina în nodul preferential;