

SDA - Arbori Minimi de Acoperire

Daniel Chiş - 2021, UPB, ACS, An I, Seria AC



Algoritmi Greedy

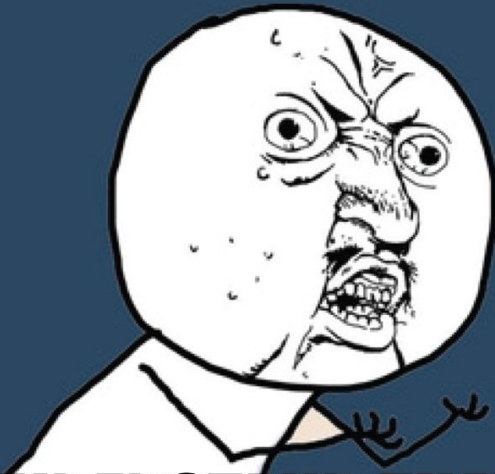


Algoritmi Greedy

Un algoritm, prin natura sa, este realizat pentru a rezolva o problemă într-o manieră optimă. Plecând de la numele lor, greedy (lacomi), acești algoritmi caută soluția cea mai apropiată.

Algoritmii Greedy caută mereu un optim local, care să ducă la un optim global. Totuși, asta nu asigură, că mereu o să fie optimi și global.

GREEDY ALGORITHM



**Y U NO UNDERSTAND THAT 'GREED
LEADS TO GRIEF'**



Arbori Minimi de Acoperire

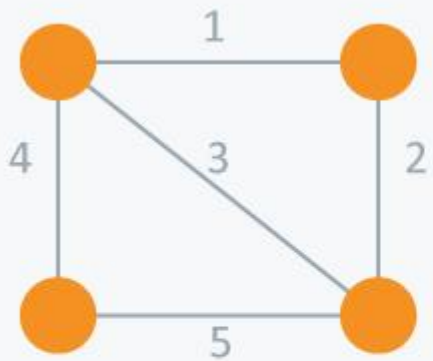
Introducere

Arbore de acoperire (spanning tree)

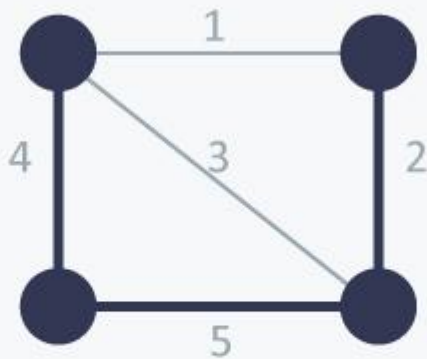
Se dă un graf neorientat și conectat $G=(N,M)$. Un arbore de acoperire peste G este un arbore care acoperă toate nodurile lui G , reprezentând în același timp și un subgraf al lui G (toate muchiile din arbore se regăsesc și în graful inițial).

Arbore minim de acoperire (minimum spanning tree)

Costul unui arbore de acoperire reprezintă suma valorilor fiecărei muchii din arbore. Într-un arbore pot exista mai mulți arbori de acoperire, arborele minim de acoperire fiind cel care are cel mai mic cost.

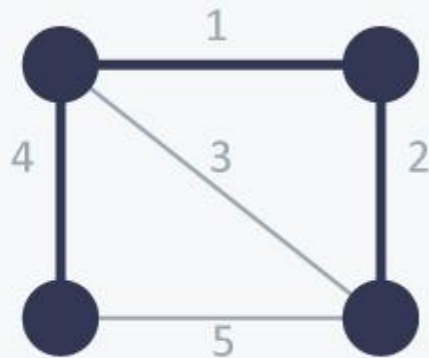


Undirected
Graph



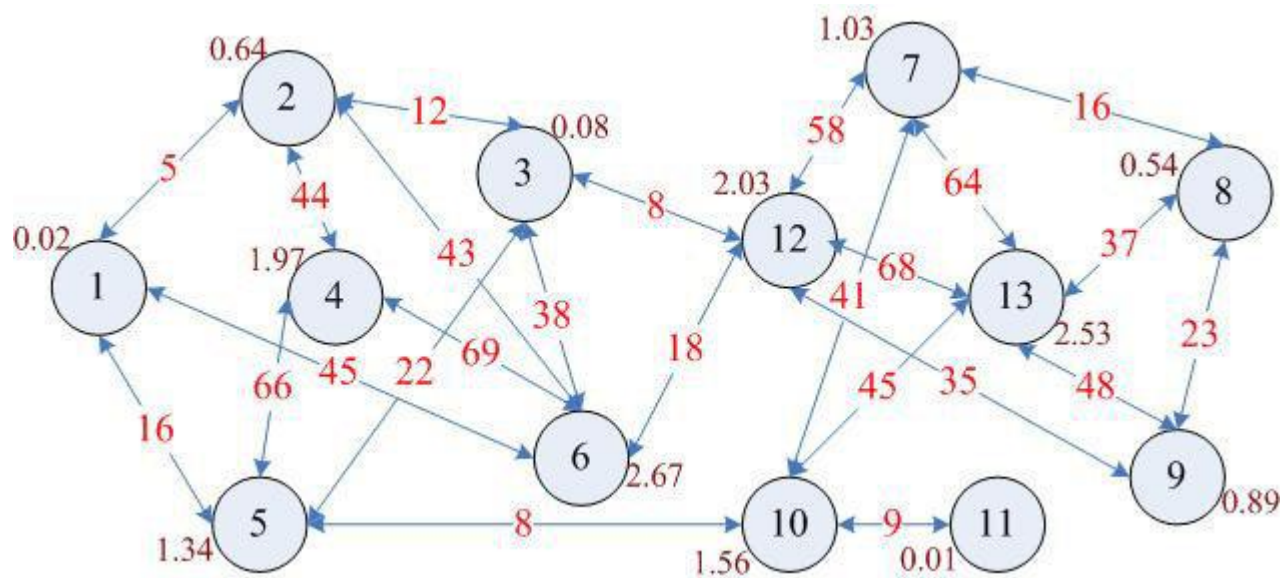
Spanning
Tree

Cost = $11 (= 4 + 5 + 2)$



Minimum Spanning
Tree

Cost = $7 (= 4 + 1 + 2)$



Exemplu graf cu muchii și costuri asociate

Aplicații

Arborii minimi de cost sunt foarte utilizați în rezolvarea problemelor de cost ce implică grafurile. Exemple:

- Rețele de internet
- Analiză de clustere
- Recunoașterea scrisului de mână
- Segmentări de imagini

Algoritmul lui Kruskal

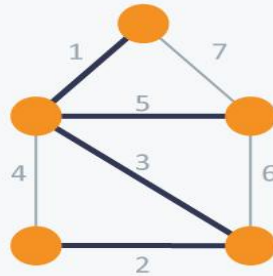
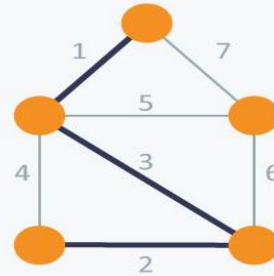
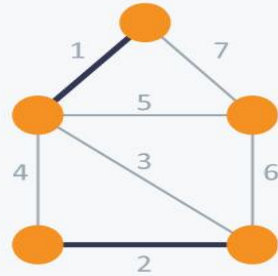
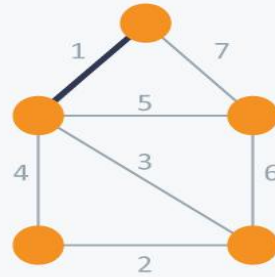
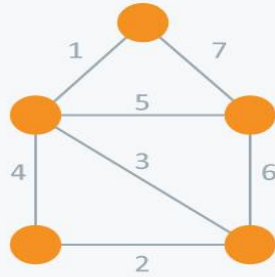
Joseph Kruskal a fost un matematician, statistician și computer scientist din Statele Unite.

Algoritmul lui Kruskal este unul folosit pentru aflarea arborelui de cost minim pentru un graf. În acest algoritm muchiile sunt adăugate la arborele de acoperire în ordinea crescătoare a valorilor. Dacă o muchie adăugată formează un graf, atunci ea este eliminată.

Algoritm:

- Se sortează muchiile din graf crescător în funcție de costul acestora
- Se adaugă muchiile la noul arbore creat
- Se adaugă doar muchiile care nu formează un graf

Kruskal's Algorithm



Algoritmul lui Prim

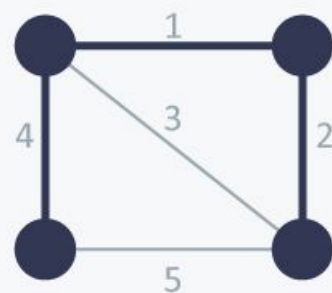
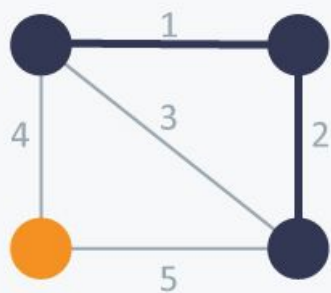
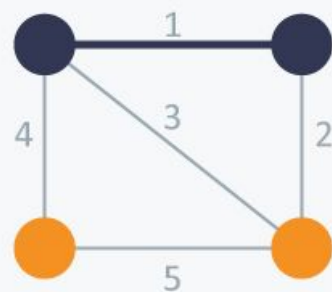
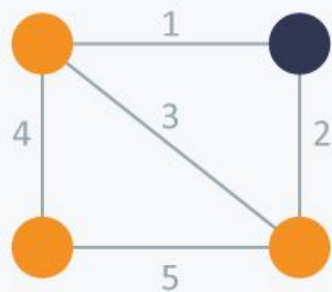
Creat de Vojtěch Jarník, matematician ceh.

Algoritmul lui Prim este folosit pentru a găsi arborele de cost minim. Cu acest algoritm creăm arborele plecând de la o poziție de start aleasă. Spre deosebire de algoritmul lui Kruskal, unde ne folosim de muchii, aici ne folosim de noduri.

Algoritm:

- Se crează o listă cu toate muchiile din graf și costurile asociate
- Se crează un schelet al arborelui de acoperire folosind nodurile
- Se caută muchia cu cel mai mic cost și se adaugă la scheletul de arbore, după care o ștergem din listă
- Se adaugă alte muchii având grijă ca un capăt al muchiei să fie deja în arborele creat și costul e minim

Prim's Algorithm





Exerciții

Exerciții

Creați un graf cu minim 12 noduri și 12 muchii. Fiecare muchie va avea un cost asociat.

- a) Realizați arborele minim de acoperire folosind algoritmul lui Kruskal. Afișați ordinea nodurilor, valoarea muchiilor din arbore și costul total. 4.5
- b) Realizați arborele minim de acoperire folosind algoritmul lui Prim. Afișați ordinea nodurilor, valoarea muchiilor din arbore și costul total. 4.5
- c) Datele de intrare trebuie luate dintr-un fișier de tip csv.
- d) Outputul algoritmilor trebuie afișat pe ecran și exportat în fișiere de tip csv.
- e) Exportați într-un fișier de tip text ce diferențe și asemănări incluzând beneficii și problematici pentru fiecare algoritm (o analiză comparativă)

Tema trebuie să includă și schema grafului plus toate fișierele.