**Arbore partial de cost minin**

**Problema din viata reala:**

Se dau n orase precum si costul conectarii anumitor perechi de orase. Se cere sa se aleaga o parte din muchii astfel incat se asigure existenta unui lant intre oricare doua orase si costul total sa fie minim.

**G=(V, E)**un graf neorientat, unde V are n elemente (n noduri) si E are m elemente (m  muchii).   Se cunosc costurile muchiilor. Prin eliminarea unor muchii se obtine un graf partial. Daca acesta este arbore se va numi arbore partial.

Costul unui arbore este egal cu suma costurilor muchiilor sale.

**Arborele partial de cost minim este un arbore pentru care costurl este minim.**

**Problema.** Fie un graf conex pentru care se cunosc costurile muchiilor. Sa se determine arborele de cost minim asociat (costul arborelui si vectorul de tati).

**Alg lui Prim**

**Intrare**: Un graf conex ponderat cu nodurile V și muchiile E.

**Initializare**:

Vnou = {x}, unde x este un nod arbitrar (punct de plecare) din V,

Enou= {}

**repetă până când Vnou=V:**

- alege muchia (u,v) din E de cost minim astfel încât u este în Vnou și v nu e

(dacă există mai multe astfel de muchii, se alege arbitrar)

- se adaugă v la Vnou, (u,v) la Enou

**Ieșire**: Vnou și Enou descriu arborele parțial de cost minim

**Algoritmul lui Kruskal** este un algoritm în teoria grafurilor care găsește arborele parțial de cost minim pentru un graf conex ponderat. Cu alte cuvinte, găsește submulțimea muchiilor care formează un arbore care include toate vârfurile și care este minimizat din punct de vedere al costului. Dacă graful nu este conex, atunci algoritmul găsește o pădure parțială de cost minim (un arbore parțial de cost minim pentru fiecare componentă conexă). Algoritmul lui Kruskal este un exemplu de algoritm greedy.

Algoritmul funcționează în felul următor:

* creează o pădure F (o mulțime de arbori), unde fiecare vârf din graf este un arbore separat
* creează o mulțime S care conține toate muchiile din graf

**atât timp cât S este nevidă**

- elimină o muchie de cost minim din S

- **dacă** acea muchie conectează doi arbori distincți,

**atunci** adaugă muchia în pădure, combinând cei doi arbori

într-unul singur

**altfel**, ignoră muchia

La sfârșitul algoritmului, pădurea are doar o componentă care reprezintă un arbore parțial de cost minim al grafului.

**Problema propusa:**

Pentru graful de mai jos determinati arborele partial de cost minim prin aplicarea manuala a algoritmului lui Prim sau Kruskal.

