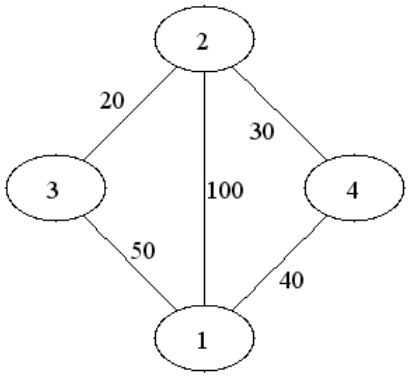
Algorithme de Kruskal
(recherche d'un arbre couvrant de poids minimal)

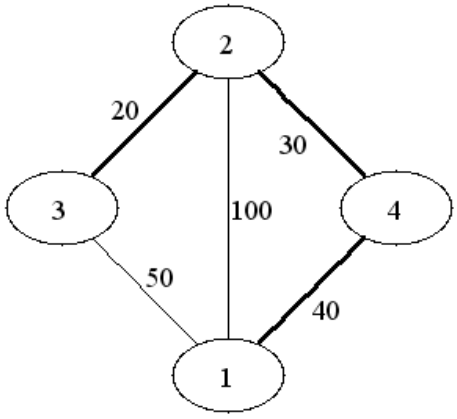
On souhaite équiper les différents sites d'une entreprise d'un réseau local. Pour ce faire, on doit poser des câbles. On connaît les points entre lesquels des câbles peuvent être posés. Ces points sont numérotés 1…N. On connaît aussi le coût de pose d'un câble. On peut représenter toutes ces données par le dessin suivant :



Poser une câble entre les points 1 et 2 coûte 100, entre 2 et 4 coûte 30, … Le fait que 3 et 4 ne sont pas reliés exprime qu'on ne pourra pas poser de câble 3-4 parce qu'il coûte trop cher ou parce qu'il n'est pas faisable techniquement, …

Le problème qu'on se pose est le suivant : Quels câbles doit-on effectivement poser pour que :

1- tous les points soient reliés par le même réseau
2- le coût total des câbles posés soit minimal



La solution représentée ci-dessus (câbles sélectionnés en gras) répond bien aux points 1. et 2. Elle coûte 70. Il n'y a pas moyen de faire mieux.

Les dessins ci-dessus sont des graphes. En terminologie de graphes, les points s'appellent des sommets et les câbles des arêtes. Le problème général posé est celui de l'arbre couvrant de poids minimal. Un des algorithmes bien connus pour résoudre ce problème est l'algorithme de Kruskal. Toutes ces définitions ainsi que l'algorithme de Kruskal seront vus en cours.

La structure du programme à écrire est la suivante :

a) Lire un graphe à partir d'un fichier texte
b) Trier les arêtes du graphe par l'algorithme du tri rapide, dans l'ordre croissant de leur coût
c) Appliquer l'algorithme de Kruskal
d) Ecrire le résultat dans un format lisible par le logiciel de visualisation de graphes Graphviz

1- Représentation interne du graphe :

- Vous devez utiliser une classe `arete` dont les variables d'instances sont 3 entiers : les deux extrémités de l'arête et son coût.
- L'ensemble des arêtes du graphe seront stockées dans un tableau d'arêtes (donc de type `arete[]`)
- Un graphe sera donc représenté par la paire : nombre de sommets, tableau d'arêtes

2- Format d'entrée

Le fichier d'entrée aura le format suivant :

```
Nombre de sommets
Nombre d'arêtes
Pour chaque arête : x y v
Où x et y sont les extrémités de l'arête et v est son coût
```

Ainsi pour le graphe donné en exemple, on aura le fichier :

```
4
5
1 2 100
2 3 20
1 4 40
1 3 50
2 4 30
```

3- Format de sortie

Comme indiqué ci-dessus, la sortie se fera dans un fichier lisible par Graphviz. Il vous sera a priori suffisant de savoir que le premier dessin de ce document est le résultat de la lecture par Graphviz du fichier suivant :

```
graph G {
"1" -- "2" [label=100];
"2" -- "3" [label=20];
"2" -- "4" [label=30];
"1" -- "4" [label=40];
"1" -- "3" [label=50];
}
```

Dans un premier temps, vous pouvez réaliser les étapes a) et b) du programme. Des précisions vous seront données plus tard, en particulier pour l'étape c).