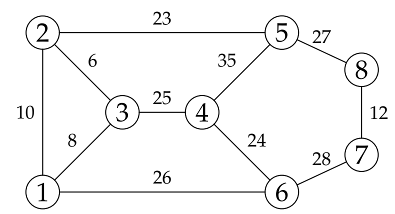
Colle n^{o} 3 d'Informatique – CaML
Arbre couvrant minimal d'un graphe

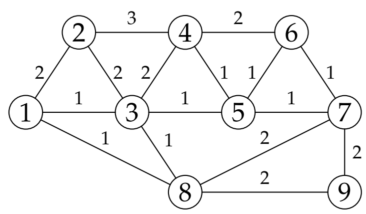
Mathieu Goutelle (Mathieu.Goutelle@ens-lyon.fr)

24 – 28 novembre 2003

1 Définitions

Un *graphe valué* est un triplet (S, A, π) avec S un ensemble fini de *sommets*, A un ensemble d'*arêtes*. Une arête est définie comme un couple (x, y) de sommets distincts. $\pi : A \to \mathbb{R}$ est une application telle que $\pi(x, y)$ est le *poids* de l'arête (x, y). Les figures suivantes donnent deux exemples de graphe valué.



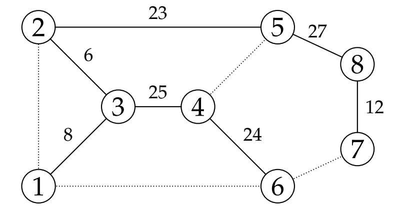


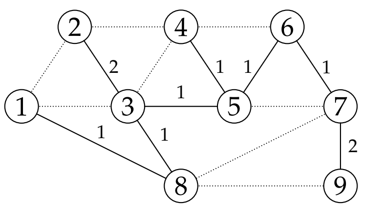
Les notions de *voisin*, de *chemin*, de *cycle* et de *connexité* sont supposées connues. Un graphe est *acyclique* s'il ne comporte aucun cycle. Un chemin est dit *élémentaire* ou *injectif* s'il ne repasse pas deux fois par un même sommet. **Tous les graphes considérés dans la suite sont connexes et non-orientés.** On note $s = |S|$ le nombre de sommets et $a = |A|$ le nombre d'arêtes. $G' = (S', A')$ est un *sous-graphe* de $G = (S, A)$ si $S' \subset S$ et $A' \subset A$. C'est un *graphe partiel* si de plus $S' = S$. Soit S' une partie de S et A' l'ensemble des arêtes de G reliant deux sommets appartenant à S' ; nous dirons que (S', A') est le sous-graphe de G *induit par* S'.

Soit $G = (S, A)$ un graphe. On rappelle l'équivalence des définitions d'*arbre* suivantes :

i. G est connexe et acyclique ;
ii. G est connexe et $a = s - 1$;
iii. G est acyclique et $a = s - 1$;
iv. G est acyclique, mais ne le reste pas si on lui retire une arête ;
v. G est connexe, mais ne le reste pas si on lui retire une arête ;
vi. deux sommets quelconques de G sont reliés par un unique chemin élémentaire.

Un *arbre couvrant* de G est un graphe partiel de G connexe et acyclique. L'arbre est *minimal* si la somme des poids de ses arêtes est minimale. Une *forêt* est une famille d'arbres deux à deux disjoints. La définition d'une *forêt recouvrante* est intuitive.





2 Quelques remarques d'ordre général

D'abord, on peut supposer, sans perte de généralité, que tous les poids sont positifs ou nuls (**Pourquoi ?**).

Il est assez facile de montrer l'existence d'un arbre couvrant. L'existence d'un minimum est également évidente puisque l'on travaille sur un ensemble fini d'arbres couvrants. Par contre, l'unicité n'est pas vérifiée dans le cas général : **donner un contre-exemple où un graphe admet deux arbres couvrants minimaux**. L'unicité est vérifiée lorsque les poids des arêtes sont deux à deux disjoints.

Pour terminer ces remarques, expliquer pourquoi la stratégie « diviser pour régner » suivante échoue pour construire un arbre couvrant minimal : si le graphe n'a qu'un sommet, le problème est résolu. Sinon, soient $S = S_1 \cup S_2$ une partition de l'ensemble des sommets, T_1 un arbre recouvrant minimal du sousgraphe de G induit par S_1 et T_2 défini de façon analogue. Parmi les arêtes reliant un sommet de S_1 et un sommet de S_2, choisissonsen une de poids minimal. En réunissant T_1, T_2 et cette arête, on obtient un arbre recouvrant minimal du graphe G.

Dans la suite, vous utiliserez les deux types CaML suivants pour réaliser l'implantation des algorithmes proposés :

```
type arete = Edge of int*int*int;;
type graphe = { Sommets: int list; Aretes: arete list };;
```

3 Algorithme de Prim

R.C. PRIM, *Shortest connection networks and some generalizations*,
in Bell System Tech J. 36 (1957). pages 1389-1401.

L'algorithme de Prim est le suivant. On note P l'ensemble des sommets déjà traités et O l'ensemble des arêtes du graphe partiel que l'on est en train de construire.

1. Choisir au hasard un sommet ξ du graphe : $P \leftarrow \xi$;
2. Chercher l'arête $\alpha = (x, y)$ de poids minimal reliant un élément $x \in P$ et un élement $y \in S \setminus P$;
3. Recommencer à l'étape 2 avec $P \leftarrow P \cup \{y\}$ et $O \leftarrow O \cup \{\alpha\}$ jusqu'à ce que $P = S$. Le cas échéant, O est un arbre couvrant minimal de G.

Implémenter cet algorithme en CaML.

4 Algorithme de Kruskal

J.B. Kruskal *On the shortest spanning subtree of a graph and traveling salesman problem*,
in Proceedings of the American Mathematical Society 7 (1956). pages 48-50.

L'algorithme de Kruskal est le suivant. On note F une forêt de sous-graphes disjoints de G.

1. Initialement, F contient l'ensemble des sous-arbres élémentaires de G, *i.e.* l'ensemble des arbres à un seul nœud ;
2. Chercher l'arête α de poids minimal reliant deux éléments x et y de F ;
3. Remplacer dans F x et y par leur réunion en utilisant l'arête α ;
4. Recommencer à l'étape 2 jusqu'à ce que F soit réduit à un seul élément qui est un arbre couvrant minimal de G.

Implémenter cet algorithme en CaML.

Pour information, ce thème a été « concourisé » l'année dernière aux CCP, plus particulièrement l'algorithme de Prim. Si vous souhaitez traiter des détails supplémentaires, n'hésitez à y jetter un coup d'œil...