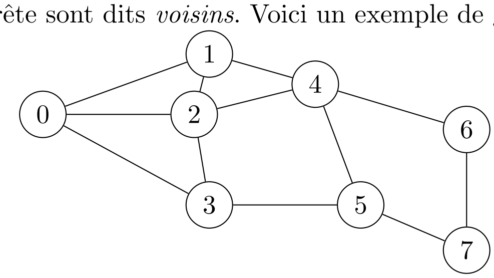
TP 5: Algorithmes sur des graphes

7 février 2014

Dans ce TP, nous allons travailler sur des graphes. On découvrira principalement l'algorithme de Dijkstra et l'algorithme de Prim qui permettent de calculer efficacement respectivement le plus court chemin entre deux sommets, et un arbre couvrant de poids minimal.

1 Introduction

Un graphe non orienté G est la donnée d'un ensemble de sommets S et d'un ensemble d'arêtes $A \subseteq Set_2(S)$ liant deux sommets entre eux, où $Set_2(S)$ est l'ensemble des sous-ensembles de S à deux éléments. Deux sommets liés par une arête sont dits *voisins*. Voici un exemple de graphe :



avec $S = \{0, 1, 2, 3, 4, 5, 6, 7\}$ et $A = \{\{0, 1\}, \{0, 2\}, \{0, 3\}, \{1, 2\}, \{2, 3\}, \{1, 4\}, \{2, 4\}, \{3, 5\}, \{4, 5\}, \{4, 6\}, \{5, 7\}, \{6, 7\}\}$.

En machine, on choisira de représenter un graphe par ses listes d'adjcacence, c'est-à-dire un tableau tel que la i-ième entrée du tableau liste tous les voisins du sommet i. Voici les listes d'adjacence de l'exemple : $[[1, 2, 3], [0, 2, 4], [0, 1, 3, 4], [0, 2, 5], [1, 2, 5, 6], [3, 4, 7], [4, 7], [5, 6]]$.

Le *degré d'un sommet* est son nombre de voisins. Le *degré d'un graphe* est le degré du sommet de plus haut degré. Un *chemin* dans un graphe est une suite de sommets $v_1, v_2, \ldots, v_k$ telle que $\{v_i, v_{i+1}\}$ est dans A pour tout $i \in \{1, \ldots, k-1\}$. C'est un *cycle* si $v_1 = v_k$. Un chemin est dit *élémentaire* s'il ne contient pas de cycle. La *longueur* d'un chemin élémentaire $v_1, v_2, \ldots, v_k$ est le nombre d'arêtes empruntées $k - 1$. La *distance* entre deux sommets u et v, notée $d(u, v)$; est la longueur minimale d'un chemin entre u et v. Dans l'exemple, 0,1,4,5,7 est un chemin élémentaire de longueur 4, 0,2,4,5,3,0 est un cycle, et $d(2, 7) = 3$. Un graphe est dit *connexe* si toute paire de sommets est reliée par un chemin. Un graphe est un *arbre* si c'est un graphe connexe sans cycle.

Exercice 1 *Écrire la fonction degré, qui prend en entrée un graphe représenté par ses listes d'adjacences, et qui retourne son degré.*

Exercice 2 *Écrire une fonction qui prend en entrée un graphe et un de ses sommets v et retourne la liste de tous les sommets accessibles depuis v.*

Exercice 3 *Écrire une fonction qui dit si un graphe est connexe ou non.*

Exercice 4 *Écrire une fonction qui dit si un graphe est un arbre ou non.*

Exercice 5 *Écrire une fonction qui prend en entrée un graphe et deux de ses sommets u et v et qui retourne $d(u, v)$.*

2 Plus courts chemins

Trouver un plus court chemin dans un graphe a de multiples applications (GPS, routage...). Un graphe pondéré est un graphe dont chaque arête est associée à un *poids*, qui sera, ici, toujours un entier strictement positif. On adapte naturellement la notion de longueur d'un chemin comme la somme des poids des arêtes empruntées.

Exercice 6 *Réfléchir en groupe à un algorithme qui calcule le plus court chemin entre deux sommets d'un graphe.*

2.1 Algorithme de Dijkstra

Exercice 7 *Implémenter l'algorithme de Dijkstra (présenté au tableau).*

Exercice 8 *Montrer la correction de l'algorithme.*

2.2 Algorithme de Floyd-Warshall

Maintenant, on s'intéresse à la table de tous les plus courts chemin (entre chaque paire de sommets).

Exercice 9 *Implémenter l'algorithme de Floyd-Warshall (présenté au tableau).*

3 Arbres couvrants

Un sous-graphe $G = (S', A')$ d'un graphe $G = (S, A)$ est un graphe tel que $S' \subseteq S$ et $A' \subseteq A$. Un arbre est un graphe connexe sans cycle. Un arbre couvrant d'un graphe $G = (S, A)$ est un sous-graphe $G = (S, A)$ de G tel que G' est un arbre. Le poids d'un arbre couvrant est la somme des poids de ses arêtes. On s'intéresse au problème suivant : étant donné un graphe G, trouver un arbre de poids minimal.

Exercice 10 *Implémenter l'algorithme de Prim (présenté au tableau).*

Exercice 11 *Montrer la correction de l'algorithme.*