Algorithmique : TD 10
Licence M/I - 2 décembre 2004

1 Distances dans un graphe

Soit $G = (S, A)$ un graphe orienté dont chaque arête e a un poids $w(e)$ strictement positif. Le poids d'un chemin est la somme des poids de ses arêtes et la distance entre deux sommet est le poids minimal des chemins entre ces deux sommets. On veut calculer pour tout couple de sommets p et q la distance entre eux dans G. Pour cela, on utilise une matrice M telle que $M(p, q)$ contient le poids de l'arête (p, q).

Exercice 1: Comment calculer le poids des chemins de longueur 2 à partir de M, idem pour les chemins de longueur 3.

Exercice 2: Montrer que tout chemin de poids minimal entre p et q est un chemin sans circuit.

Exercice 3: En déduire un algorithme pour calculer la distance entre tout couple de sommet. Quelle est sa complexité.

Exercice 4: Comment calculer directement le poids des chemins de longueur 4 à partir de ceux de longueur 2 ? En déduire un algorithme et sa complexité.

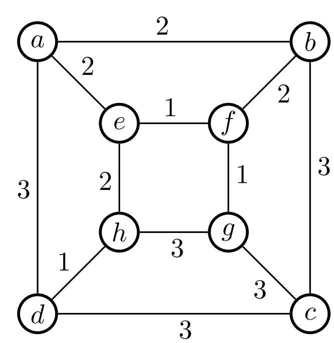
2 Arbres couvrants de poids minimal

Soit G un graphe connexe. Un arbre couvrant G est un sous-graphe de G, sans cycle et connexe, qui contient tous les sommets de G.

On suppose que chaque arête a un poids $w(i, j)$ positif. Le but de l'algorithme qui suit est de calculer un arbre couvrant de G de poids total (somme des poids des arêtes) minimal. On dira qu'une arête est adjacente à un sous-graphe si elle est adjacente à un de ses sommets et qu'elle ne fait pas partie du sous-graphe. D'autre part, quand on ajoute une arête à un graphe, on ajoute évidemment aussi (si besoin est) les sommets correspondants.

Algorithme de Prim :
Soit s un sommet et $ACM \leftarrow s$
Tant que ACM ne recouvre pas G faire
    Soit a une arête adjacente à ACM de poids minimum qui ne crée pas de cycle
    $ACM \leftarrow ACM \cup \{a\}$
Retourner ACM

Exercice 5: Appliquer cet algorithme sur le graphe suivant :



Exercice 6: Quel nombre d'arêtes comporte un arbre couvrant d'un graphe à n sommets ?

Exercice 7: Comment peut-on vérifier en temps constant, au cours de l'algorithme, que l'arête a que l'on considère n'appartient pas à ACM et ne va pas créer de cycle ?

Exercice 8: Quelle structure de données peut-on utiliser pour stocker les arêtes adjacentes à ACM ? Comment doit-on stocker le graphe pour que l'algorithme soit efficace ?

3 Questions supplémentaires

Distances, suite : On numérote les sommets de 1 à $|S|$. On notera p_i le sommet numéroté i. A l'étape n, on effectue les boucles suivantes :
Pour $p \in S$ faire
  pour $q \in S$ faire
    $d \leftarrow w(p, p_n) + w(p_n, q)$
    si $d < w(p, q)$ alors $w(p, q) \leftarrow d$

Exercice 9: Montrer qu'après l'étape n, $w(p, q)$ contient le poids minimal des chemins de p à q dont tous les sommets intérieurs ont un numéro inférieur ou égal à n. En déduire qu'après l'étape $|S|$, $w(p, q)$ est la distance entre p est q. Quelle est la complexité de cet algorithme ?

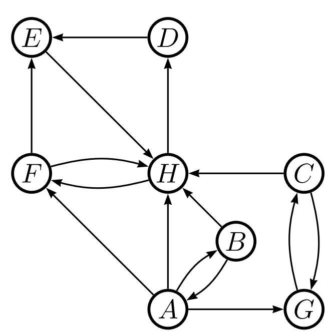
Arbre couvrant, suite : Écrire l'algorithme de Prim en détaillant chaque opération algorithmique (c'est-à-dire en tenant compte des questions précédentes). Quelle est la complexité de cet algorithme ?

On va maintenant montrer que l'algorithme de Prim est correct, c'est-à-dire qu'il calcule effectivement un arbre couvrant de poids minimal. La preuve est par récurrence sur le nombre de sommets traités. On montre qu'à chaque étape, ACM est inclus dans un arbre couvrant minimal T.

Exercice 10: Montrer la propriété est-elle vraie au début de l'algorithme. On suppose qu'à un moment donné, ACM est inclus dans un arbre couvrant minimal T. Montrer que ACM' obtenu en ajoutant une arête adjacente de poids minimal qui ne crée pas de cycle est contenu dans un arbre couvrant minimal T' (on pourra examiner comment construire un tel arbre à partir de T).

Autres questions :

On considère le graphe suivant, dans lequel les sommets sont rangés par ordre alphabétique :



Exercice 11: Dessiner l'arborescence correspondant à un parcours en profondeur à partir de A. Donner l'ordre topologique induit par ce parcours. Calculer, à l'aide de l'algorithme de Tarjan les composantes fortement connexes du graphe.