

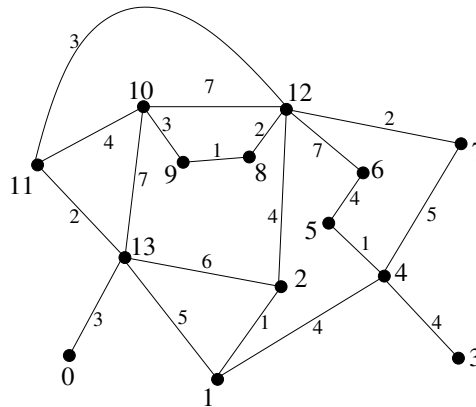
info602 : théorie des graphes et algorithmes sur les graphes

TD 4 : arbres couvrants optimaux

Pierre Hyvernat
Laboratoire de mathématiques de l'université de Savoie
bâtiment Chablais, bureau 22
téléphone : 04 79 75 94 22
email : Pierre.Hyvernat@univ-savoie.fr
www : <http://www.lama.univ-savoie.fr/~hyvernat/>

Exercice 1 : algorithme de Kruskal

Question 1 : en détaillant bien toutes les étapes, appliquez l'algorithme de Kruskal sur le graphe suivant.



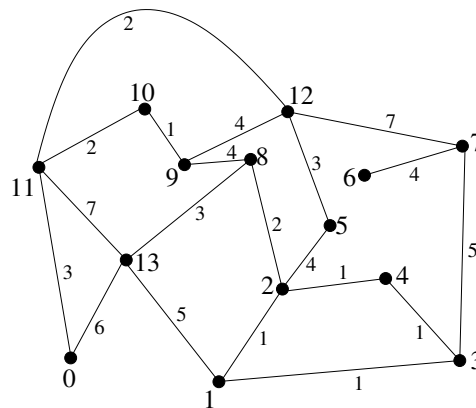
Quel est le poids d'un arbre couvrant de poids minimal ?

Question 2 : en utilisant le même graphe que la question précédente, calculer le poids d'un arbre couvrant de poids maximum. Donner un arbre de poids couvrant maximal.

Exercice 2 : algorithme de Prim

Question 1 : discuter des différences fondamentales entre l'algorithme de Kruskal et l'algorithme de Prim. Comment résumer ces différences ?

Question 2 : appliquez l'algorithme de Prim vu en cours pour calculer un arbre couvrant de poids minimal du graphe suivant



Question 3 : un tas binaire est une structure de données en arbre binaire, où le père est toujours plus petit que ces deux fils. De plus, on impose que l'arbre soit bien équilibré (les branches maximales sont de taille t ou $t + 1$). Expliquez comment améliorer la complexité de l'algorithme du cours en utilisant des tas binaires.

À quelle complexité estimez-vous le nouvel algorithme ?

Exercice 3 : labyrinthes

Question 1 : essayez d'adapter les algorithmes de Prim et de Kruskal à la génération de labyrinthes sur des quadrillages.

Question 2 : quelle méthode proposez-vous pour résoudre un labyrinthe ?

Question 3 : implantez un algorithme de création de labyrinthe dans un langage de programmation de votre choix.

Exercice 4 : ça marche

Question 1 : soit G un graphe connexe non-orienté, et p une fonction de poids sur ses arêtes. Soit T un arbre couvrant (pas forcément de poids minimal) ; si e est une arête de T , on note $\omega_T(e)$ l'ensemble d'arêtes suivant :

$$\{xy \in A_G \mid x \in T_1, y \in T_2\}$$

où T_1 et T_2 sont les deux composantes connexes de $T \setminus \{e\}$. Cet ensemble est appelé "cocycle fondamental de T associé à e ".

En reprenant l'arbre de poids minimal trouvé dans l'exercice 1, question 1, dessinez quelques cocycles fondamentaux.

Question 2 : montrez la chose suivante

Propriété. Avec les notations de la question précédente, on a : T est de poids minimal si et seulement si, pour toute arête e de T et toute arête f de $\omega_T(e)$, on a $p(e) \leq p(f)$.

Pour la direction \Leftarrow , il faudra considérer un arbre T^* de poids minimal avec le plus possible d'arêtes en commun avec T . Si T^* a les mêmes arêtes que T , c'est terminé ; sinon, on considère une arête e dans T mais pas dans T^* , ...

Question 3 : si f est une arête de G qui n'est pas dans T , on appelle "cycle fondamental de T associé à f " l'unique cycle de $T \cup \{f\}$. On le note $\mu_T(f)$. Montrez la propriété suivante :

Propriété. T est de poids minimal si et seulement si, pour toute arête f en dehors de T et toute arête e de $\mu_T(f)$, on a $p(e) \leq p(f)$.

Pour la direction \Leftarrow , utilisez la question précédente.

Question 4 : déduisez que les algorithmes de Kruskal et de Prim fonctionnent correctement. (càd qu'ils donnent bien un arbre couvrant de poids minimal.)