Université Paris Diderot
L3 Informatique

Algorithmique
Année 2008-2009, 1^{er} semestre

TD n°8

Arbres couvrants de poids minimum.

Dans ce TD, $G = (V, E)$ est un graphe fini, non-orienté, sans boucle, dont les arêtes sont pondérées. La fonction $\omega : E \to \mathbb{R}^+$ est la fonction de poids des arêtes. Le poids de G, noté $\omega(G)$, est la somme des (R(E(E ω

Données : $G = (V, E)$ un graphe, ω une fonction de pondération des arêtes.
Résultat : $T = (V, A)$ un ACPM de G.

$S \leftarrow v \in V$
tant que $S \neq V$ **faire**
    $e \leftarrow min(\delta(S))$ où $e = (x, y)$ avec $x \in S$
    $A \leftarrow A \cup \{e\}$
    $S \leftarrow S \cup \{y\}$
retourner $T = (V, A)$

Algorithme 1 : Algorithme de Prim

1. Procédez à une exécution de l'algorithme de Prim sur l'exemple de l'exercice 1.
2. Quelle structure de donnée doit-on employer pour implémenter cet algorithme ?
3. Quelle est la complexité de l'algorithme ?
4. Montrez que l'algorithme de Prim, dans un modèle comparatif, est équivalent au tri.
5. Que peut-on conclure de la borne inférieure de complexité pour l'algorithme de Prim ?

Exercice 4 [Arbre couvrant de poids max.] On se propose maintenant d'étudier le problème de trouver l'arbre couvrant T de poids maximum : T est toujours couvrant mais cette fois pour tout arbre couvrant T' de G, $\omega(T) \geqslant \omega(T')$.

Est-ce que ce problème est relié à l'ACPM ?

Donnez un algorithme pour trouver un tel arbre. Quelle est sa complexité ?

Exercice 5 [Arbre couvrant de poids min, $\times$.] On se propose maintenant d'étudier le problème de trouver l'arbre couvrant de poids minimum quand la fonction de valuation est définie par :

$$\omega(G) = \prod_{(x,y)\in E(G)} \omega(x, y)$$

Est-ce que ce problème est relié à l'ACPM ?

Donnez un algorithme pour trouver un tel arbre. Quelle est sa complexité ?

Exercice 6 [Algorithme de Kruskal]

Données : $G = (V, E)$ un graphe, ω une fonction de pondération des arêtes.
Résultat : $T = (V, A)$ un ACPM de G.

1 $S \leftarrow \emptyset$
2 $A \leftarrow \emptyset$
3 $ET \leftarrow$ L'ensemble des arêtes de E triées de manière croissante
4 **tant que** $S \neq V$ **faire**
5     $e = (x, y) \leftarrow$ **RetirerEnTete**(ET)
6   **si** *e ne crée pas de cycle dans* $T = (S, A)$ **alors**
7         $S \leftarrow S \cup \{x\} \cup \{y\}$
8         $A \leftarrow A \cup \{e\}$
9 **retourner** $T = (S, A)$

Algorithme 2 : Algorithme de Kruskal

Procédez à une exécution de l'algorithme de Kruskal sur l'exemple de l'exercice 1.

En supposant que le test de la ligne 6 se fait en $f(n, m)$, où f est une fonction inconnue, calculer la complexité de l'algorithme en fonction de n, m et f.