Université Paris Diderot – Paris 7
L3 Informatique

Algorithmique
Année Semaine du 31 octobre 2011

TD n°6

Parcours de graphes

Exercice 1 Appliquer au graphe de la figure 1 l'algorithme de parcours en largeur (le sommet origine est indiqué par une simple flèche entrante). L'arbre de parcours en largeur résultant sera présenté par un schéma dans lequel les sommets de profondeur égale seront mis à la même hauteur, le sommet origine étant mis en haut.

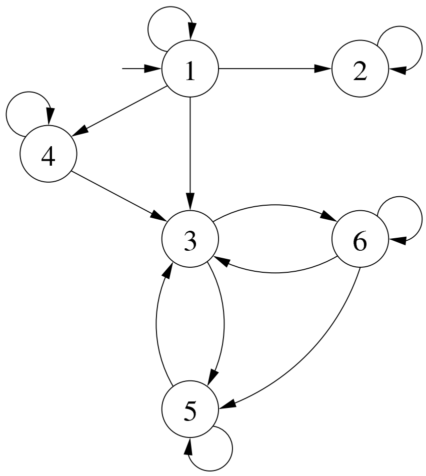

FIGURE 1 – Un graphe orienté

Exercice 2

Étant donnés un graphe orienté $G = (S, A)$ et un sommet $s \in S$ appartenant à ce graphe, on appelle *arborescence des plus courts chemins de G à partir de s* un sous-graphe $T = (S_T, A_T)$ de G tel que

- T est un arbre,
- S_T est l'ensemble des sommets de G accessibles depuis s (y compris s lui-même),
- et pour tout sommet $x \in S_T$, le chemin de s à x dans T est un plus court chemin de s à x dans G.

Étant donnée une telle arborescence, est-il toujours possible de l'obtenir par un certain parcours en largeur de G depuis s ? Prouvez votre affirmation.

Exercice 3 Appliquer au graphe de la figure 1 l'algorithme de parcours en profondeur en nommant pour chaque arc du graphe son type : arc du parcours, arc avant, arc transverse ou arc retour.

Exercice 4 Proposer un algorithme renvoyant, s'il existe, un circuit de longueur minimale passant par un sommet s donné dans un graphe G. Prouver sa correction.

Exercice 5 Proposer un algorithme qui vérifie si un graphe *non orienté* et connexe possède un cycle. Prouver sa correction.

Exercice 6 Modifier l'algorithme de parcours en profondeur donné en cours de telle sorte qu'il affiche pour chaque arc du graphe son type (arc du parcours, arc retour, arc transverse, arc avant).