Université Bordeaux 1 Licence semestre 6 – Année 2006-2007

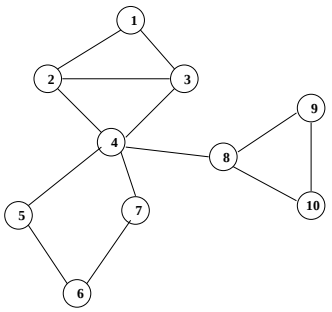
Algorithmique des Graphes – TD 2 et 3

Exercice 1

```
PP(G)
1  pour chaque sommet u de X
2      faire couleur[u] <- BLANC
3            pi[u] <- nil
4  temps <- 0
5  pour chaque sommet u de X
6      faire si couleur[u] = BLANC
7          alors Visiter_PP(u)
```

```
Visiter_PP(u)
1  couleur[u] <- GRIS
2  d[u] <- temps <- temps + 1
3  pour chaque v de Adj[u]
4      faire si couleur[v] = BLANC
5          alors pi[v] <- u
6                Visiter_PP(v)
7  couleur[u] <- NOIR
8  f[u] <- temps <- temps+1
```

Appliquer l'algorithme de parcours en profondeur PP au graphe G1 de la figure ci-dessous :



Donner aussi la classification des arêtes (*de l'arbre*, *retour*, *avant*, *transversaux*). (on conviendra que dans les listes d'adjacence les sommets sont rangés dans l'ordre croissant de leur numéro.)

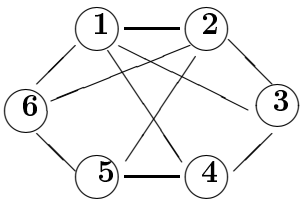
Exercice 2

1. Soit G un graphe non orienté, et soit x un de ses sommets. Ajouter à l'algorithme PP les instructions nécessaires pour compter le nombre de sommets de la composante connexe de x. Est-ce que l'on peut obtenir le même resultat en utilisant le parcours en largeur ?

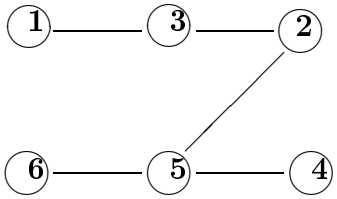
2. Ajouter à l'algorithme PP les instructions nécessaires pour tester si un graphe non orienté possède un cycle.

Exercice 3

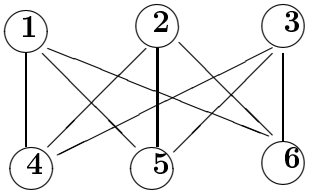
On considère les deux graphes non orientés $H1$ et $H2$ et les deux arbres $T1$ et $T2$ ci-dessous :



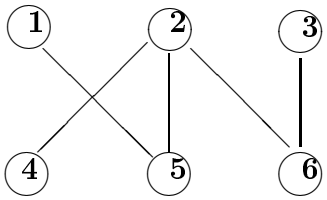
H1



T1



H2



T2

Si $G = (X, A)$ est un graphe non orienté ayant ensemble de sommets X et ensemble d'arêtes A, un arbre *recouvrant* de G est donné par un sous ensemble de l'ensemble des arêtes A formant un arbre ayant X pour ensemble de sommets. Pour posséder un arbre recouvrant, un graphe doit être connexe.

Est-ce que $T1$ et $T2$ peuvent respectivement être des arbres recouvrants de $H1$ et $H2$ obtenus par un parcours en profondeur ? Justifier les réponses en spécifiant l'éventuel sommet de départ et l'ordre des sommets dans les différentes représentations du graphe.

Même question pour le parcours en largeur.

Exercice 4

Mise en œuvre d'un algorithme pour vérifier la forte connexité d'un graphe orienté `G` :

1. Appliquer à `G` l'algorithme de parcours en largeur à partir d'un sommet x_0 : qu'est-ce qu'on peut en déduire (au sujet de la forte connexité) si à la fin du parcours il y a des sommets pas encore visités ? Et si tous les sommets ont été visités ?

2. Construire le graphe `G'` obtenu de `G` en enversant tous les arcs. Appliquer à `G'` l'algorithme de parcours en largeur à partir de x_0 : qu'est-ce qu'on peut en déduire (au sujet de la forte connexité) si à la fin du parcours il y a des sommets pas encore visités ? Et si tous les sommets ont été visités ?

Justifier vos réponses et proposer un algorithme pour constuire `G'`.

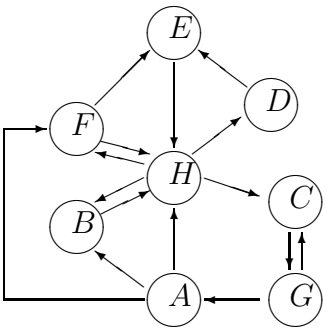
Peut-on envisager d'utiliser l'algorithme de parcours en profondeur à la place de l'algorithme de parcours en largeur ?

Exercice 5

1. Modifier l'algorithme PP pour calculer la longueur du plus long chemin dans un graphe orienté sans cycles. Comment effectuer ce même calcul en utilisant l'algorithme PL ? Comparer les compléxités.
2. Même question pour un graphe non orienté sans cycles.

Exercice 6

On considère le graphe $\mathcal{G}$ suivant :



On suppose que les sommets sont rangés par ordre alphabétique, ainsi que les successeurs d'un sommet.

1. Donner l'ordre de visite des sommets dans un parcours en largeur de $\mathcal{G}$ à partir du sommet A.
2. Caractériser les arcs de $\mathcal{G}$ (de l'arbre, retour, avant, transversaux) dans le parcours en profondeur à partir du sommet A.
3. Quelles sont les composantes fortement connexes du graphe $\mathcal{G}$?
4. Si l'on supprime du graphe $\mathcal{G}$ l'arc $G \longrightarrow A$, en quoi cela modifie-t-il les réponses aux questions précédentes ?