

DS - Algorithmique 2 - Master 1 Bioinformatique

1 avril 2015 - Durée : 1h30 minutes

Les documents, les ordinateurs et les téléphones ne sont pas autorisés.

Exercice 1

On suppose que vous disposez d'un module python implémentant les arbres binaires et qui contient les fonctions suivantes :

```
def racine(A)
    # renvoie la racine de l'arbre A
def pere(A,p)
    # renvoie le père du sommet p dans l'arbre A ou None
    # si c'est la racine
def fils_gauche(A, p)
    # renvoie le fils gauche du sommet p dans l'arbre A
def fils_droit(A, p)
    # renvoie le fils droit du sommet p dans l'arbre A
```

On dit qu'un arbre est équilibré si, pour chaque sommet s , la différence entre la hauteur du sous-arbre droit de s et la hauteur du sous-arbre gauche de s vaut -1 , 0 ou 1 . Écrire, en python, la fonction `est_equilibre(arbre)` qui renvoie `True` si l'arbre passé en paramètre est équilibré et `False` sinon.

Exercice 2

Appliquer l'algorithme du *parcours en largeur* $PL(G, s)$ au graphe G_1 à partir du sommet s_0 (on conviendra que dans les listes d'adjacence, les sommets sont rangés dans l'ordre lexicographique). Donner l'ordre d'entrée des sommets dans la file. Pour chaque sommet visité, indiquer la valeur de `pere[s]` et de `d[s]`. L'algorithme $PL(G, s)$ est redonné en annexe.

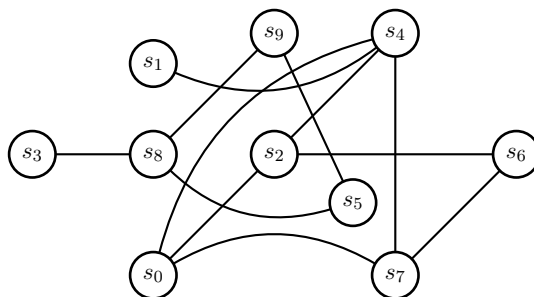


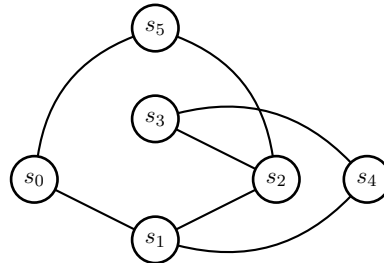
FIGURE 1 – Graphe G_1

Exercice 3

1. Rapeller la définition de composante connexe d'un graphe. Quels sont les composantes connexes de G_1 .
2. Donner un algorithme qui détermine le nombre de composantes connexes d'un graphe non orienté.
3. Peut-on utiliser votre algorithme pour compter le nombre de composantes connexes d'un graphe orienté ? Pourquoi ?

Exercice 4

1. L'excentricité d'un sommet v d'un graphe non orienté $G = (V, E)$, notée $exc(v)$, est la distance maximum entre v et les autres sommets du graphe (∞ si G n'est pas connexe). Modifier l'algorithme de parcours en largeur pour calculer l'excentricité d'un sommet et appliquer votre algorithme au sommet s_0 du graphe ci-dessous.



2. Le diamètre d'un graphe non orienté G est la plus grande distance entre deux sommets, autrement dit le maximum des excentricités.
 - Montrer que si n est le nombre de sommets, il est suffisant de calculer l'excentricité de $n - 1$ sommets.
 - Donner un exemple de graphe et l'ordre des sommets utilisé pour lequel il est nécessaire de calculer les $n - 1$ premières excentricités.
 - Donner l'algorithme de calcul du diamètre d'un graphe.

Rappel :

```
0  PL(G,s)
1  pour chaque sommet u de X[G] \ {s} faire
2      couleur[u] <- BLANC
3      d[u] <- infini
4      pere[u] <- nil
5  couleur[s] <- GRIS
6  d[s] <- 0
7  pere[s] <- nil
8  Enfiler(F, s)
9  tant que non vide(F) faire
10     u <- tete(F)
11     pour chaque v de Adj(u) faire
12         si couleur[v] = BLANC
13             alors couleur[v] <- GRIS
14                 d[v] <- d[u] + 1
15                 pere[v] <- u
16                 Enfiler(F, v)
17     Defiler(F)
18     couleur[u] <- NOIR
```