

DS - Algorithmique - Master 1 Bio-informatique - session 2

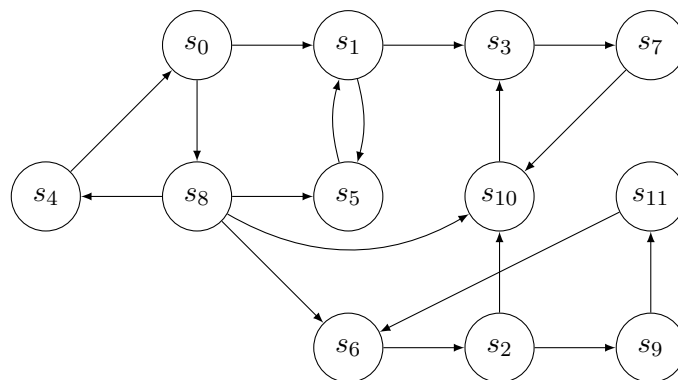
juin 2017 - Durée : 1h30 minutes

Les documents sont autorisés.

Nous supposons que les graphes sont sans arêtes multiples.

Exercice 1

Appliquer l'algorithme de parcours en profondeur PP au graphe de la figure ci-dessous (on conviendra que dans les listes d'adjacence les sommets sont rangés dans l'ordre croissant de leur numéro). Vous donnerez l'arbre d'exécution de PP et `VisiterPP` ainsi que les valeurs de d , f et $pere$ de chaque sommet.



Rappel :

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

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

Exercice 2

Soit G un graphe de dépendance. Un graphe de dépendance est un graphe orienté dont :

- les sommets représentent des tâches à réaliser (par exemple, pour un spectacle, ces tâches pourraient être : monter un chapiteau, monter les gradins, installer l'orchestre, monter l'estrade, demander des autorisations à la mairie). On notera ces tâches par les lettres de l'alphabet A, B, C, D, \dots

- les arcs représentent les dépendances entre ces tâches. C'est à dire que si (A, B) est un arc du graphe, alors cela veut dire que la réalisation de la tâche A dépend de la réalisation de la tâche B . Ainsi, pour faire la tâche A il faut impérativement faire la tâche B avant.

Par exemple, si la tâche A représente la tâche "cuire le gâteau" et la tâche E représente la tâche "préparer la pâte", alors il y aura, dans le graphe de dépendances, un arc (A, E) . En effet, pour cuire un gâteau il faut préparer sa pâte avant.

1. Existe-t-il des graphes où il n'est pas possible de réaliser toutes les tâches décrites par le graphe de dépendances ? Si oui, donner un critère qui permet de déterminer ces graphes impossibles et illustrez-le sur un exemple.
2. Soit G un graphe de dépendances réalisables. Donnez un algorithme qui permet de donner l'ordre dans lequel il faut réaliser les tâches de G .
Vous illustrerez votre algorithme sur un exemple non trivial.
3. Proposer un algorithme qui prend en paramètre un graphe de dépendances G et qui retourne VRAI si les tâches du graphe G peuvent toutes être réalisées et FAUX sinon. Vous illustrerez votre algorithme sur un exemple non trivial.

Exercice 3

Proposer un algorithme qui, pour tout graphe non orienté G , retourne VRAI si et seulement si G possède un sommet (ou plus) de degré 1, dans les deux cas suivants :

1. G est représenté par une *matrice d'adjacence* A ,
2. G est représenté par une *liste d'adjacence* Adj , et
3. G est représenté par une *matrice d'incidence* B .