

Éléments de correction du devoir surveillé du 16 novembre 2016

Dans tout le sujet on a convenu que, dans les différentes structures de données, les sommets sont rangés dans l'ordre croissant de leur numéro.

Exercice 1

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

1. G est représenté par *matrice d'adjacence* A ,
2. G est représenté par *listes de successeurs* Adj

Dans les deux cas, donner la complexité de votre solution en la justifiant.

Solution :

1. Matrice d'adjacence $A[i,j]$:

```
pour i allant de 1 à n faire:
  k = 0
  pour j allant de 1 à n faire:
    si A[i,j]==1 alors:
      k = k+1
      si k==2 alors:
        sortir // sort de la boucle la plus interne
      fin si
    fin si
  fin pour
  si k==1 alors:
    retourner VRAI
  fin si
fin pour
retourner FAUX
```

Complexité : $\mathcal{O}(n^2)$ car dans le pire des cas il faut parcourir entièrement la matrice et pour chaque élément les instructions à effectuer sont en temps constant.

2. listes de successeurs Adj :

```
pour chaque sommet u de V(G) faire:
  k = 0
  pour chaque sommet v de Adj[u] faire:
    k = k+1
    si k==2 alors:
      sortir // sort de la boucle la plus interne
    fin si
  fin pour
  si k==1 alors:
    retourner VRAI
  fin si
fin pour
retourner FAUX
```

Complexité : $\mathcal{O}(n)$ car pour chaque sommet on visite au maximum deux successeurs en exécutant des instructions (tests, affectations) en temps constant.

Exercice 2

L'excentricité d'un sommet v d'un graphe non orienté $G = (V, E)$, notée $exc(v)$, est le maximum des distances entre v et les autres sommets du graphe (∞ si G n'est pas connexe).

1. Modifier l'algorithme de **parcours en largeur** pour calculer l'excentricité d'un sommet et appliquer votre algorithme au sommet s_0 du graphe G_1 ci-dessous..
2. Quelle est la complexité de votre algorithme ?

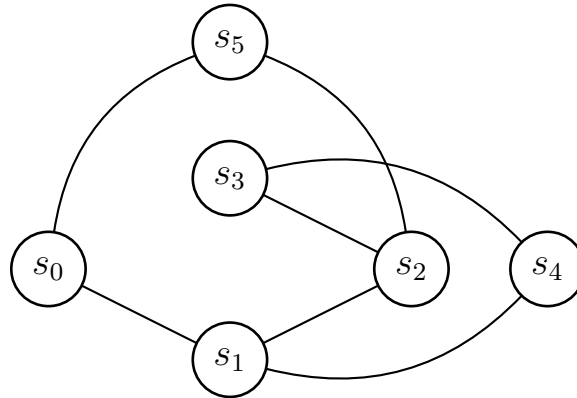


FIGURE 1 – Graphe G_1

Solution :

1. Dans l'algorithme de parcours en largeur donné en annexe ajouter :
 ligne 7' :

```
compteur <- 0 // pour compter les sommets visités
```


 ligne 14' : (la distance de chaque sommet est supérieure ou égale à celle des sommets qui ont été dans la file avant lui)

```
distanceMax <- d[v]
```



```
compteur <- compteur+1
```


 ligne 19 (après la boucle tant que) :

```
si compteur < |X(G)| alors distanceMax <-  $\infty$  // si graphe non connexe
```



```
retourner distanceMax
```
2. Complexité : $\mathcal{O}(m + n)$ c'est à dire la même que le parcours en largeur car on a ajouté au parcours des instructions en temps constant.

3. Soit G un graphe connexe. Montrer que G contient un cycle impair (un cycle ayant un nombre impair d'arêtes) si et seulement si quelque soit $u \in V(G)$ il existent deux sommets v_1 et v_2 adjacents tels que

$$\text{dist}(u, v_1) = \text{dist}(u, v_2).$$

Solution :

1. Par définition, il existe un chemin C_{uw} entre u et w de longueur $\text{dist}(u, w)$ et un chemin C_{wv} de longueur $\text{dist}(w, v)$. L'union de ces deux chemins est un chemin entre u et v (pas forcément élémentaire, il peut emprunter certaines arêtes plusieurs fois) de longueur $\text{dist}(u, w) + \text{dist}(w, v)$. La distance $\text{dist}(u, v)$ étant la longueur du chemin le plus court entre u et v , elle est donc inférieure ou égale à la longueur du chemin entre u et v passant par w , d'où l'inégalité demandée.
2. Puisque v_1 et v_2 sont adjacents, on a $\text{dist}(v_1, v_2) = 1$. D'après la question précédente, en posant $w = v_1$ et $v = v_2$, on obtient

$$\text{dist}(u, v_2) \leq \text{dist}(u, v_1) + 1.$$

Par symétrie, (en posant $w = v_2$ et $v = v_1$) on obtient

$$\text{dist}(u, v_1) \leq \text{dist}(u, v_2) + 1.$$

D'où

$$-1 \leq \text{dist}(u, v_1) - \text{dist}(u, v_2) \leq 1.$$

3. Montrons d'abord (par l'absurde) que si G contient un cycle impair, alors quelque soit $u \in V(G)$ il existe $v_1, v_2 \in V(G)$ adjacents tels que $\text{dist}(u, v_1) = \text{dist}(u, v_2)$.

Supposons qu'un graphe connexe contienne un cycle impair, mais que la conclusion ne soit pas vraie, c-à-d, qu'il existe $u \in V(G)$ tel que pour tout couple de sommets adjacents $v_1, v_2 \in V(G)$ on a $\text{dist}(u, v_1) \neq \text{dist}(u, v_2)$. D'après la question précédente, $|\text{dist}(u, v_1) - \text{dist}(u, v_2)| = 1$, donc si $\text{dist}(u, v_1)$ est paire, alors $\text{dist}(u, v_2)$ est impaire et vice versa. Autrement dit, pour tout couple de sommets adjacents, il y en a toujours un (et un seul) à distance de u paire et un (et un seul) à distance de u impaire.

Considérons maintenant un cycle impair C dans G . Il y a un certain nombre de sommets appartenant à C à distance de u paire, les autres sont à distance de u impaire. Or, d'après l'observation précédente, ces deux ensembles de sommets doivent s'alterner le long du cycle, ce qui n'est pas possible tant que la longueur du cycle est impair. Absurde!

Montrons maintenant que si quelque soit $u \in V(G)$ il existent $v_1, v_2 \in V(G)$ tels que $v_1 v_2 \in E(G)$ et $\text{dist}(u, v_1) = \text{dist}(u, v_2)$, alors G contient un cycle impair.

Fixons un sommet u quelconque. D'après la supposition, il existe un couple de sommets v_1, v_2 adjacents tels que $\text{dist}(u, v_1) = \text{dist}(u, v_2)$. Considérons l'arborescence A de parcours en largeur à partir du sommet u . Dans A , il existe un chemin C_1 de liaison entre u et v_1 de longueur $\text{dist}(u, v_1)$ dans A ; il existe un autre chemin C_2 de liaison entre u et v_2 de la même longueur. Si on enlève de l'union de ces deux chemin-là la partie commune éventuelle, on obtiendra un chemin de longueur paire (on enlève la même partie de deux chemins de la même longueur) entre v_1 et v_2 . (Observer que puisque A est un arbre, le chemin élémentaire entre v_1 et v_2 est unique!). L'arête reliant v_1 et v_2 et chemin de longueur paire entre v_1 et v_2 forment un cycle de longueur impaire dans G , ce que l'on cherche à démontrer.