

Les exercices marqués par (*) sont à traiter en travail personnel.

Parcours en largeur

Exercice 1

Appliquer l'algorithme du *parcours en largeur* $PL(G, s)$ au graphe non-orienté G_1 de la feuille 1 à partir du sommet s_1 et au graphe orienté G_1 de la feuille 2 à partir du sommet s_2 . Chaque fois donner l'ordre d'entrée des sommets dans la file, et le contenu final des tableaux $d[]$ et $pere[]$.

```
0  PL(G,s)
1  pour chaque sommet u de V[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
```

Exercice 2

(*) Proposer une modification de l'algorithme $PL(G, s)$ permettant de détecter si un graphe (non-orienté) donné est connexe ou pas. Spécifier les lignes du code ci-dessus à modifier / supprimer / ajouter. Étudier la complexité de votre algorithme.

Exercice 3

Proposer une modification de l'algorithme $PL(G, s)$ permettant de détecter si un graphe (non-orienté) donné est un arbre ou pas. Spécifier les lignes du code-dessus à modifier / supprimer / ajouter. Étudier la complexité de votre algorithme.

Exercice 4

(*) Dans un graphe non orienté, on définit $\text{dist}(u, v)$, distance de u à v , comme étant le nombre minimum d'arêtes d'une chaîne reliant les sommets u et v (ou ∞ s'il n'existe aucune chaîne de u à v).

1. Soit G un graphe connexe, soient $u, v, w \in V(G)$. Montrer que

$$\text{dist}(u, v) \leq \text{dist}(u, w) + \text{dist}(w, v).$$

2. Soit G un graphe connexe, soient $u, v_1, v_2 \in V(G)$. Montrer que si $v_1 v_2 \in E(G)$, alors

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

3. Soit G un graphe connexe. Montrer que G contient un cycle impair (un cycle possédant un nombre impair d'arêtes) si et seulement si quelque soit $u \in V(G)$ il existe $v_1, v_2 \in V(G)$ tels que $v_1 v_2 \in E(G)$ et

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

Exercice 5

Un graphe (non orienté) est *biparti* si l'on peut partager l'ensemble de ses sommets en deux sous-ensembles tels qu'aucune arête du graphe relie deux sommets appartenant au même sous-ensemble.

1. Montrer que si un graphe est biparti, alors il ne peut pas contenir de cycle impair.
2. Soit G un graphe biparti. Soit u et v deux sommets de G reliés par une chaîne. Montrer que u et v sont dans la même partition si et seulement si la longueur de la chaîne les reliant est paire.
3. En s'appuyant sur les énoncés précédents et l'exercice précédent, proposer un algorithme permettant de déterminer si un graphe est biparti.

Exercice 6

(*) Proposer un algorithme qui, en faisant deux appels à la fonction $\text{PL}(G, s)$, détermine si un graphe orienté donné est fortement connexe ou pas.

Solutions des exercices optionnels

Correction de l'exercice 2

$PL(G, s)$ ne visite que les sommets de la composante connexe de s . Ainsi, compter le nombre de sommets rencontrés pendant le parcours en largeur revient à compter le nombre de sommets n_s associé à la composante connexe de s . Pour déterminer si le graphe G est connexe, il suffit de vérifier que n_s est égal au nombre de sommets n du graphe. Pour cela, on peut utiliser un compteur c de la façon suivante (ici, l'incrément est mis en ligne 15' afin qu'il ne soit exécuté qu'une seule fois par sommet) :

```
5'. c <- 1
15'. c <- c+1
18. retourner c == n
```

On a ajouté au programme PL des instructions en temps constant, donc sa complexité reste en $\mathcal{O}(n + m)$.

Remarques :

- Pour déterminer si le graphe est connexe, il n'est pas nécessaire d'utiliser les tableaux d et $pere$. Le code de leur mise à jour peut être retiré de l'algorithme.
- Si lors du parcours, on renseigne le tableau $pere$, alors on peut se passer du compteur c . Il suffit de vérifier, à la fin, que le tableau $pere$ ne contient qu'une seule occurrence de la valeur nil ;
- De façon analogue, si lors du parcours, on renseigne le tableau d , alors on peut se passer du compteur c . Il suffit de vérifier, à la fin, que le tableau d ne contient aucune occurrence de la valeur infini.

Pour résumer : pour vérifier qu'un graphe est connexe pendant le parcours en largeur, un seul tableau est nécessaire. Il s'agit du tableau $couleur$. Ensuite, on utilise au choix, un compteur, le tableau d ou bien le tableau $pere$.

Dans tous les cas la complexité du parcours reste inchangée.

Correction de l'exercice 4

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, ni simple, il peut emprunter certains sommets ou 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 si la longueur du cycle est impaire. Absurde !

Montrons maintenant que si quelque soit $u \in V(G)$ il existe $v_1, v_2 \in V(G)$ 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 obtenue par un 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 $\text{dist}(u, v_1)$ et il existe un autre chemin C_2 de liaison entre u et v_2 de même longueur. Si on enlève de l'union de ces deux chemins la partie commune éventuelle, on obtiendra un chemin de longueur paire (on enlève la même partie à 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 le chemin de longueur paire entre v_1 et v_2 forment un cycle de longueur impaire dans G , ce que l'on voulait démontrer.

Correction de l'exercice 6

Un graphe orienté est fortement connexe si pour tout couple de sommets u et v de $V[G]$ il existe un chemin de u à v et un chemin de v à u .

Soit G le graphe orienté donné. On effectue un premier parcours en largeur de G en prenant comme origine un sommet s quelconque. Pendant le parcours on comptabilisera les sommets accessibles depuis s (en incrémentant de 1 un compteur - initialisé à 1, car s est accessible depuis lui même - chaque fois qu'un nouveau sommet passe à GRIS, comme dans l'exercice 2). À la fin de ce premier parcours le compteur contient le nombre de sommets accessibles depuis s . Si ce nombre est différent de la taille de G , l'algorithme s'arrête avec réponse négative : G n'est pas fortement connexe.

Sinon on construit un deuxième graphe G' obtenu à partir de l'ensemble des sommets de G en inversant tous les arcs de G . Le deuxième parcours en largeur s'effectue sur G' toujours avec origine le sommet s . Aussi, pendant ce deuxième parcours on comptabilisera les sommets accessibles depuis s .

Si à la fin de l'appel $\text{PL}(G', s)$ la valeur du compteur coïncide avec la taille de G' , l'algorithme se termine avec réponse positive, sinon avec réponse négative.

Justification : Lors du premier parcours en largeur on vérifie qu'il existe un chemin de s vers tout autre sommet v de G . Si après avoir inversé les arcs de G (construction de G') il existe encore un chemin de s à tout autre sommet v dans G' , cela signifie que dans G il existe un chemin de v à s . Si dans G il y a un chemin dans les deux sens pour tout couple de sommets la propriété de forte connexité est vérifiée.