

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 à 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 $\text{PL}(G, \mathbf{s})$, détermine si un graphe orienté donné est fortement connexe ou pas.