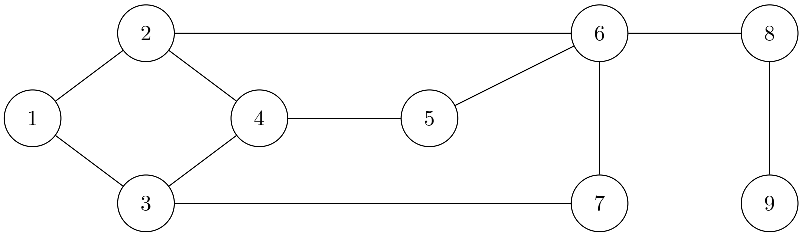
Module LU2IN003 Graphes : parcours en largeur semaine 10

Alix Munier Kordon et Maryse Pelletier

Exercice(s)

Exercice 1 – Parcours en largeur d'un graphe non orienté

On considère le graphe non orienté $G_1 = (V_1, E_1)$:



Question 1

Pour chacun des parcours génériques de G_1 suivants, dire s'il est ou non un parcours en largeur : $(8, 6, 9, 2, 5, 7, 1, 4, 3)$, $(6, 8, 5, 7, 4, 2, 3, 1, 9)$, $(4, 2, 3, 5, 1, 7, 6, 8, 9)$, $(3, 1, 4, 2, 6, 5, 7, 8, 9)$

Justifier les réponses négatives.

Solution:

On rappelle la définition du parcours en largeur.

Soit $G = (V, E)$ un graphe non orienté connexe et $L = (v_1, \ldots, v_n)$ un parcours de G d'origine v_1.

L est un *parcours en largeur* si pour tout sous-parcours $L_k = (v_1, \ldots, v_k)$ avec $k < n$, v_{k+1} est un sommet adjacent du premier sommet ouvert de L_k.

$(8, 6, 9, 2, 5, 7, 1, 4, 3)$: oui.

$(6, 8, 5, 7, 4, 2, 3, 1, 9)$: non, on n'a pas visité tous les voisins de 6 (il manque 2) avant de visiter un voisin (le sommet 4) d'un de ses voisins.

$(4, 2, 3, 5, 1, 6, 7, 8, 9)$: non, 7 n'est pas voisin du premier sommet ouvert de $(4, 2, 3, 5, 1)$ (qui est 2).

$(3, 1, 4, 2, 6, 5, 7, 8, 9)$: non, on n'a pas visité tous les voisins de 3 (il manque 7) avant de visiter un voisin (le sommet 2) d'un de ses voisins.

Question 2

Donner trois parcours en largeur de G_1, l'un partant du sommet 1, un autre du sommet 9 et un troisième du sommet 5.

Solution:

Trois, parmi d'autres :

$(1, 2, 3, 4, 6, 7, 5, 8, 9)$

$(9, 8, 6, 2, 5, 7, 1, 4, 3)$

$(5, 4, 6, 2, 3, 7, 8, 1, 9)$

Question 3

On considère le parcours $L = (3, 1, 4, 7, 2, 5, 6, 8, 9)$ de G_1. Dire quel est le premier sommet ouvert de chaque sous-parcours de L. Le parcours L est-il un parcours en largeur ?

Solution:

sous-parcours	1er sommet ouvert
(3)	3
$(3,1)$	3
$(3,1,4)$	3
$(3,1,4,7)$	1
$(3,1,4,7,2)$	4
$(3,1,4,7,2,5)$	7
$(3,1,4,7,2,5,6)$	6
$(3,1,4,7,2,5,6,8)$	8
$(3,1,4,7,2,5,6,8,9)$	–

L est un parcours en largeur : pour tout sous-parcours $L_k = (v_1, \dots, v_k)$ avec $k < 9$, v_{k+1} est bien un sommet adjacent au premier sommet ouvert de L_k.

Exercice 2 – Parcours en largeur et distance

Soit $G = (V'E)$ un graphe non orienté connexe ayant n sommets. Soit $s \in V$ et $L = (s_1, \dots, s_n)$ une liste des n sommets de G telle que $s_1 = s$ et $dist_s(s_1) \leq dist_s(s_2) \leq \dots \leq dist_s(s_n)$.

Question 1

Prouver que L est un parcours générique de G.

Solution:

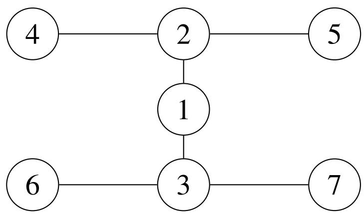
Il suffit de vérifier que tout sommet $u \neq s$ de L est adjacent à au moins un sommet situé avant lui dans L. Soit $u \neq s$ un sommet de L et soit $\mu = (u_0, \dots, u_{k-1}, u_k)$ une plus courte chaîne entre s et u, alors $dist_s(u) = k$, $dist_s(u_{k-1}) \leq k - 1$ et u est adjacent à u_{k-1}. Comme $dist_s(u_{k-1}) < dist_s(u)$, u_{k-1} est situé avant u dans L.

Question 2

L est-elle nécessairement un parcours en largeur de G ? Justifier la réponse.

Solution:

Non, par exemple :



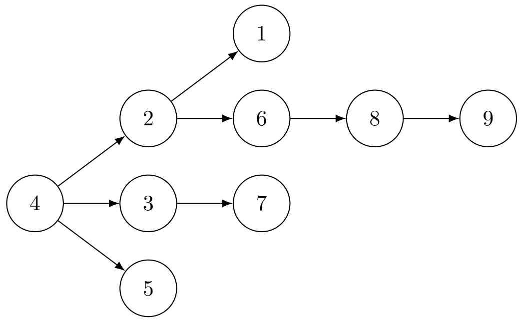
$L = (1,2,3,4,6,5,7)$ vérifie la condition de croissance des distances mais n'est pas un parcours en largeur : le plus petit sommet ouvert du sous-parcours $(1,2,3,4)$ est 2 et 6 n'est pas adjacent à 2.

Exercice 3 – Graphe de liaison en largeur

Question 1

On considère le parcours en largeur $L = (4,2,3,5,1,6,7,8,9)$ du graphe G_1. Dessiner le graphe de liaison en largeur de L. Existe-t-il un autre graphe de liaison en largeur pour L ?

Solution:



Il n'y a pas d'autre graphe de liaison en largeur pour L.

Question 2

Soit $G = (V, E)$ un graphe non orienté connexe et L un parcours en largeur de G. Montrer par contradiction que le graphe de liaison en largeur de L est unique.

Solution:

Soit $L = (s_1, \ldots, s_n)$ un parcours en largeur. Supposons qu'il existe deux graphes de liaison en largeur T et T'. Alors il existe un sommet s de G tel que le prédécesseur de s dans T est différent du prédécesseur de s dans T'. Soit i tel que $s = s_i$. Ce n'est pas possible car le prédécesseur de s dans T est le premier sommet encore ouvert de $(s_1, \ldots, s_{i-1})$ et le prédécesseur de s dans T' est aussi le premier sommet encore ouvert de $(s_1, \ldots, s_{i-1})$. Donc le prédécesseur de s dans T est égal au prédécesseur de s dans T'. Contradiction. T est donc égal à T'.

Exercice 4

On rappelle l'algorithme de construction d'un parcours en largeur :

Require: Un graphe non orienté connexe $G = (V, E)$, un sommet s
Ensure: Un parcours en largeur L d'origine s, les valeurs $dist_s(u), u \in V$

```
for all u ∈ V do
  dist_s(u) := +∞
end for
L := (), Enfiler (F,s), dist_s(s) := 0
while not FileVide(F) do
  u :=Défiler(F), L := L + (u)
  for all {u, v} ∈ E do
    if dist_s(v) = +∞ then
      Enfiler(F,v), dist_s(v) = dist_s(u) + 1
    end if
  end for
end while
```

Question 1

Appliquer cet algorithme au graphe G_1 en partant du sommet $s = 4$. Préciser, à chaque itération, le sommet u retiré à F, le sous-parcours L, la valeur de la file F, la valeur des $dist_s(v)$ pour $v \in V_1$. Les valeurs de L, de F et de $dist_s$ sont celles obtenues à chaque itération en fin du corps de boucle.

Solution:

itération	sommet u	sous-parcours L	file F	$dist_s$
0	–	$()$	(4)	$[\infty, \infty, \infty, 0, \infty, \infty, \infty, \infty, \infty]$
1	4	(4)	$(2, 3, 5)$	$[\infty, 1, 1, 0, 1, \infty, \infty, \infty, \infty]$
2	2	$(4, 2)$	$(3, 5, 1, 6)$	$[2, 1, 1, 0, 1, 2, \infty, \infty, \infty]$
4	3	$(4, 2, 3)$	$(5, 1, 6, 7)$	$[2, 1, 1, 0, 1, 2, 2, \infty, \infty]$
5	5	$(4, 2, 3, 5)$	$(1, 6, 7)$	$[2, 1, 1, 0, 1, 2, 2, \infty, \infty]$
6	1	$(4, 2, 3, 5, 1)$	$(6, 7)$	$[2, 1, 1, 0, 1, 2, 2, \infty, \infty]$
7	6	$(4, 2, 3, 5, 1, 6)$	$(7, 8)$	$[2, 1, 1, 0, 1, 2, 2, 3, \infty]$
8	7	$(4, 2, 3, 5, 1, 6, 7)$	(8)	$[2, 1, 1, 0, 1, 2, 2, 3, \infty]$
9	8	$(4, 2, 3, 5, 1, 6, 7, 8)$	(9)	$[2, 1, 1, 0, 1, 2, 2, 3, 4]$
10	9	$(4, 2, 3, 5, 1, 6, 7, 8, 9)$	$()$	$[2, 1, 1, 0, 1, 2, 2, 3, 4]$

L'algorithme est terminé, il a calculé le parcours en largeur $(4, 2, 3, 5, 1, 6, 7, 8, 9)$.

Exercice 5 – Complexité du calcul d'un parcours en largeur

On considère un graphe non orienté connexe $G = (V, E)$ ayant n sommets et m arêtes. Le but de cet exercice est d'évaluer la complexité du calcul d'un parcours en largeur. On suppose que :

— L et F sont stockés dans des listes circulaires doublement chaînées ;

— les distances $dist_s$ sont stockées dans un tableau $D[1 \dots n]$;

— le graphe non orienté est représenté par une matrice sommet-sommet, une matrice sommet-arête ou des listes d'adjacences.

Question 1

Pour tout sommet $u \in V$, on note $cv(u)$ le coût du calcul des sommets adjacents à u.

1. Rappeler sans explication l'ordre de grandeur de $cv(u)$ en fonction de la représentation de G ;
2. Donnez en fonction de $cv(u)$ l'ordre de grandeur de la boucle interne de l'algorithme.

Solution:

1. Pour la matrice sommet-sommet, $cv(u)$ est en $\Theta(n)$. Pour la matrice sommet-arête, $cv(u)$ est en $\mathcal{O}(mn)$. Pour les listes d'adjacence, $cv(u)$ est en $\Theta(d(u))$.
2. Le corps de la boucle interne est en $\Theta(1)$ et la boucle est exécuté $d(u)$ fois. La compexité de la boucle interne est donc en $\Theta(d(u) + cv(u))$. On observe que, quel que soit la représentation, $cv(u)$ est plus grand que ou égal à $d(u)$. On en déduit qu'elle est en $\Theta(cv(u))$.

Question 2

En déduire la complexité de l'algorithme de calcul d'un parcours en largeur en fonction de cv, puis pour chacune des représentations des graphes non orientés.

Solution:

— Pour l'initialisation :

```
(1) Pour tout u dans V :
        D[u] = infini
(2) L := (); F := (s) ; D[s] := 0
```

L'instruction (1) est en $\Theta(n)$, l'instruction (2) en $\Theta(1)$.

— Pour la boucle principale,

```
(3) Tant que F n'est pas vide :
        (4) u := Defiler(F)
        (5) Pour tout sommet v adjacent a u :
```

```
(6) si D[v] = infini :
        Enfiler(F,v) ; D[v] = D[u] + 1
```

L'instruction (4) est en $\Theta(1)$, l'instruction (5) est en $\Theta(cv(u))$ d'après la question précédente. Chaque sommet de G entre au plus une fois dans la file F et en sort donc au plus une fois. La complexité totale est donc en $\Theta(\sum_{u \in V}(1 + cv(u)))$, donc en $\Theta(n + \sum_{u \in V} cv(u))$.

Calculons-la en fonction de la représentation de G :

— matrice sommet-sommet : $\sum_{u\in V} cv(u) = \sum_{u\in V} n = n^2$. On obtient donc du $\Theta(n^2)$;

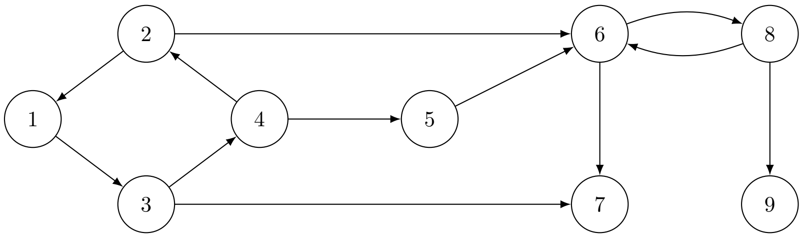
— matrice sommet-arête : $\sum_{u\in V} cv(u) = \sum_{u\in V} mn = mn^2$. On obtient donc du $\mathcal{O}(mn^2)$;

— listes d'adjacence : $\sum_{u\in V} cv(u) = \sum_{u\in V} d(u) = 2m$. On obtient donc du $\Theta(n + m)$.

Exercice 6 – Parcours en largeur des graphes orientés

Question 1

On considère le graphe orienté $G_3 = (V_3, A_3)$:



Pour chaque racine de G_3, donner un parcours en largeur de G_3.

Solution:

Pas de solution donnée.

Question 2

Adapter l'algorithme de parcours en largeur au cas des graphes orientés. L'appliquer au graphe G_3, en partant de l'une des racines.

Solution:

Pas de solution donnée.