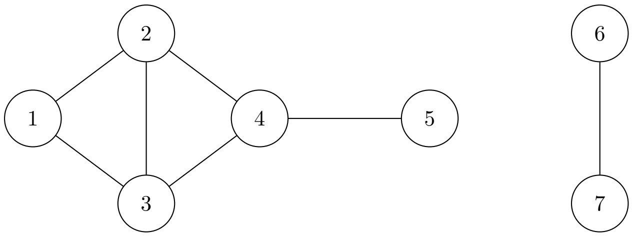
Module LU2IN003 Parcours génériques
Exercices de Base semaine 10

Alix Munier Kordon et Maryse Pelletier

## Exercice(s)

## Exercice 1 – Parcours génériques d'un graphe non orienté

On considère le graphe non orienté $G_0 = (V_0, E_0)$ :




### Question 1

Le graphe $G_0$ admet-il un parcours ? Justifier la réponse.

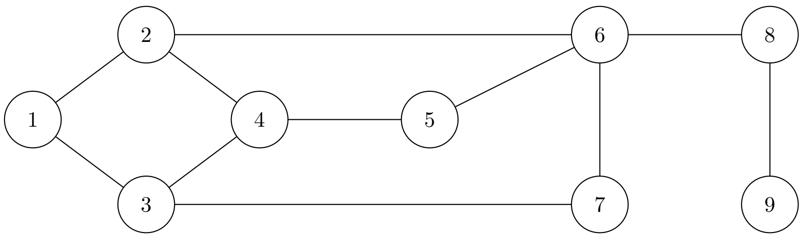
**Solution**:
Non, car $G_0$ n'est pas connexe.

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

### Question 2

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

**Solution**:
Trois, parmi bien d'autres :
$(1, 2, 3, 4, 5, 6, 7, 8, 9)$
$(9, 8, 6, 5, 7, 4, 2, 1, 3)$
$(5, 6, 2, 1, 4, 7, 3, 8, 9)$




### Question 3

La liste $(1, 2, 5, 3, 4, 6, 7, 8, 9)$ est-elle un parcours générique de $G_1$ ? Justifier la réponse.

**Solution**:
Non, car le sommet 5 est visité alors qu'il n'est voisin d'aucun sommet déjà visité.

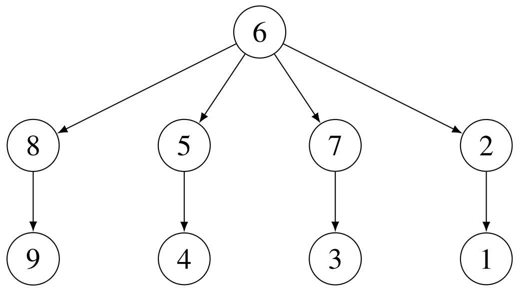
## Exercice 2 – Graphe de liaison associé à un parcours

Soit $G = (V, E)$ un graphe non orienté et $L$ un sous-parcours d'origine $s \in V$. On rappelle que $\mathcal{A}(L) = (V(L), H(L))$ est un *graphe de liaison* associé au parcours $L$ si tout sommet $v \in V(L) \setminus \{s\}$ a pour prédécesseur un sommet $u \in V(L)$ tel que $u$ est situé avant $v$ dans $L$ et $\{u, v\} \in E$.

### Question 1

On considère le graphe non orienté $G_1$ défini dans l'exercice 1. Dessiner un graphe de liaison associé au parcours $(6, 8, 5, 7, 4, 2, 3, 1, 9)$. Est-il unique ?

**Solution**:




Le graphe de liaison n'est pas unique, on pourrait choisir 4 comme prédécesseur de 2.

Une *arborescence* est un graphe orienté $G$ tel que :
– le graphe non orienté associé à $G$ est un arbre,
– $G$ possède une racine.

Un sommet $r$ est une *racine* d'un graphe orienté $G$ si, pour tout sommet $x$ de $G$, il existe un chemin de $r$ à $x$.

### Question 2

Soit $G = (V, E)$ un graphe non orienté et $L = (s_1, \ldots, s_n)$ un parcours d'origine $s_1 \in V$. Démontrer que tout graphe de liaison associé à $L$ est une arborescence de racine $s$.

**Solution**:

Soit $\mathcal{A} = (V, H)$ un graphe de liaison associé à $L$. Tout graphe de liaison est construit de la manière suivante : tout sommet $s_i \in V \setminus \{s_1\}$ a pour unique prédécesseur un sommet $s_j \in V$ tel que $s_j$ est situé avant $s_i$ dans $L$ et $\{s_j, s_i\} \in E$ (on remarque donc que $j < i$). On pose alors $p(s_i) = s_j$ et $H = \{(p(s_i), s_i), i \in \{2, \ldots, n\}\}$.

Pour $1 \leq j \leq n$, soient alors $V_j = \{s_1, \ldots, s_j\}$ et $H_j = \{(p(s_i), s_i) \mid s_i \in V_j \setminus \{s_1\}\}$. Montrons par récurrence faible sur $j$ que $\mathcal{A}_j = (V_j, H_j)$ est une arborescence de sommets $V_j$ et de racine $s = s_1$. Soit alors $A_j$ le graphe non orienté associé à $\mathcal{A}_j$ dont les sommets sont $V_j$ et les arêtes correspondent aux arcs de $H_j$.

**Base :** pour $j = 1$, $\mathcal{A}_1 = (\{1\}, \emptyset)$ est une arborescence de sommets $\{1\}$ et de racine $s$, la propriété est donc vérifiée.

**Induction :** Supposons que la propriété soit vraie pour une valeur $j \in \{1, \ldots, n-1\}$.
- Pour construire $\mathcal{A}_{j+1}$, on ajoute le sommet $s_{j+1}$ à $V_j$ et l'arc $(p(s_{j+1}), s_{j+1})$, à $H_j$. Par hypothèse de récurrence, $A_j$ est un arbre. Comme on lui ajoute un sommet $s_{j+1}$ et une arête $\{p(s_{j+1}), s_{j+1}\}$, on ne crée pas de cycle et de plus $A_{j+1}$ est connexe. $A_{j+1}$ est donc un arbre.
- Par hypothèse de récurrence, $s$ est une racine de $\mathcal{A}_j$. Par conséquent, il existe un chemin de $s$ à tout sommet $s_i$ de $V_j$ dans $\mathcal{A}_j$. Dans $\mathcal{A}_{j+1}$, il y a un arc $(p(s_{j+1}), s_{j+1})$, donc il y a un chemin (passant par $p(s_{j+1})$) de $s$ à $s_{j+1}$. On en conclut que $s$ est une racine de $\mathcal{A}_{j+1}$.

**Conclusion :** La propriété est donc vérifiée par récurrence faible.

On a montré que $\mathcal{A} = (V, H)$ est une arborescence.

## Exercice 3 – Algorithme de calcul d'un parcours d'un graphe non orienté

On rappelle l'algorithme de calcul d'un parcours vu en cours :

**Require:** Un graphe non orienté $G = (V, E)$, un sommet $s$
**Ensure:** Un parcours $L$ des sommets
  $L := (s)$, $\mathcal{B} = \mathcal{B}(L)$
  **while** $\mathcal{B} \neq \emptyset$ **do**
    Choisir un sommet $u \in \mathcal{B}$
    $L := L + (u)$
    $\mathcal{B} := \mathcal{B}(L)$
  **end while**

### Question 1

Appliquer cet algorithme au graphe $G_1$ défini dans l'exercice 1, en partant du sommet 7. Préciser, à chaque itération, le sommet $u$ choisi dans $\mathcal{B}$ (s'il y a plusieurs choix possibles, on prendra le sommet de plus petit numéro), le sous-parcours $L$, la valeur de la bordure $\mathcal{B}$ de $L$. Les valeurs de $L$ et $\mathcal{B}$ sont celles obtenues à chaque itération en fin du corps de boucle.

<u>**Solution**</u>:
Initialement : $L = (7)$, $\mathcal{B} = \{3, 6\}$
Itération 1 : $u = 3$, $L = (7, 3)$, $\mathcal{B} = \{6, 1, 4\}$
Itération 2 : $u = 1$, $L = (7, 3, 1)$, $\mathcal{B} = \{6, 4, 2\}$
Itération 3 : $u = 2$, $L = (7, 3, 1, 2)$, $\mathcal{B} = \{6, 4\}$
Itération 4 : $u = 4$, $L = (7, 3, 1, 2, 4)$, $\mathcal{B} = \{6, 5\}$
Itération 5 : $u = 5$, $L = (7, 3, 1, 2, 4, 5)$, $\mathcal{B} = \{6\}$
Itération 6 : $u = 6$, $L = (7, 3, 1, 2, 4, 5, 6)$, $\mathcal{B} = \{8\}$
Itération 7 : $u = 8$, $L = (7, 3, 1, 2, 4, 5, 6, 8)$, $\mathcal{B} = \{9\}$
Itération 8 : $u = 9$, $L = (7, 3, 1, 2, 4, 5, 6, 8, 9)$, $\mathcal{B} = \emptyset$
L'algorithme est terminé, il a calculé le parcours $(7, 3, 1, 2, 4, 5, 6, 8, 9)$.

### Question 2

À faire à la maison. Appliquer l'algorithme au graphe $G_1$ en partant d'un autre sommet ou en suivant une autre stratégie lorsque plusieurs choix sont possibles dans $\mathcal{B}$ (par exemple, prendre le sommet de plus grand numéro, ou choisir le premier sommet qui a été ajouté à $\mathcal{B}$, ou le dernier ajouté, etc.).

<u>**Solution**</u>:
Pas de solution donnée.

## Exercice 4 – Complexité du calcul d'un parcours générique

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 générique. On suppose que :

- $L$ est stocké dans une liste circulaire doublement chaînée ;
- la bordure $\mathcal{B}$ est stockée dans un tableau $B[1 \dots n]$ à valeurs dans $\{0, 1\}$ et tel que $B[u] = 1$ si $u \in \mathcal{B}$ ;
- le graphe non orienté est représenté par une matrice sommet-sommet, une matrice sommet-arête ou des listes d'adjacences. On notera par la suite $V$ une liste simplement chaînée contenant tous les voisins d'un sommet $u \in V$.

### Question 1

Décrire un algorithme qui permet de calculer la bordure $\mathcal{B}(L)$ d'un sous-parcours $L$. On pourra utiliser $V$ pour stocker les voisins d'un sommet $u \in V$.

<u>**Solution**</u>:
Au départ, $\mathcal{B} = \emptyset$. Il faut calculer, pour tout sommet $u \in L$, l'ensemble de ses voisins $V = \Gamma(u)$ pour les ajouter à la bordure si ils ne sont pas dans $L$. Cela donne en pseudo-code :

**Require:** Un graphe non orienté $G = (V, E)$, une liste $L$

**Ensure:** La bordure $\mathcal{B}(L)$
  $\mathcal{B} = \emptyset$
  **for all** $u \in L$ **do**
    $V := \Gamma(u)$
    **for all** $v \in V$ **do**
      **if** $v \notin L$ **then**
        $\mathcal{B} := \mathcal{B} \cup \{v\}$
      **end if**
    **end for**
  **end for**

## Question 2

On note $cv(u)$ la complexité pour calculer et stocker dans $V$ l'ensemble des sommets adjacents à $u$. Que vaut $cv(u)$ en fonction de la représentation de $G$ ?

<u>**Solution**</u>:

- Si $G$ est représenté par une matrice sommet-sommet, $cv(u)$ est en $\Theta(n)$.
- Si $G$ est représenté par une matrice sommet-arête, $cv(u)$ est en $\mathcal{O}(mn)$.
- Si $G$ est représenté par des listes d'adjacence, $cv(u)$ est en $\Theta(d(u))$.

## Question 3

Calculez la complexité du calcul de la bordure $\mathcal{B}(L)$ pour un sous-parcours $L$ de $k$ éléments en fonction de $cv$.

<u>**Solution**</u>:

- Pour $u \in L$ fixé, la boucle interne du calcul de la bordure est exécutée $d(u)$ fois. Tester si $v \notin L$ est en $\mathcal{O}(k)$ et l'instruction $\mathcal{B} := \mathcal{B} \cup \{v\}$ est en $\Theta(1)$. Donc la complexité de la boucle interne est en $\mathcal{O}(k \times d(u))$.
- Le corps de la boucle externe est donc en $\mathcal{O}(k \times d(u) + cv(u))$. Elle est exécutée pour tous les éléments de $L$, elle est donc en $\mathcal{O}(k^2 \times d(u) + k \times cv(u))$.

## Question 4

En déduire la complexité du calcul d'un parcours générique $L = (v_1, \ldots, v_n)$ en fonction de $cv$.

<u>**Solution**</u>:

Initialement $L$ a un élément, sa taille augmente de 1 à chaque tour de boucle et l'algorithme s'arrête lorsque $\mathcal{B} = \emptyset$, c'est-à-dire lorsque $L$ a $n$ éléments. Pour $L$ de taille $k$, il faut calculer le coût des trois instructions suivantes :

```
(1) Choisir un sommet u dans B
(2) L := L + (u)
(3) B := B(L)
```

Les instructions (1) et (2) sont en $\Theta(1)$. La troisième instruction est en $\mathcal{O}(k^2 \times d(u) + k \times cv(u))$. Si on note $L = (v_1, \ldots, v_n)$ le parcours obtenu, la complexité au total est en $\mathcal{O}(\sum_{i=1}^{n} i \times cv(v_i) + \sum_{i=1}^{n} d(u_i) \times i^2)$. Elle est donc en $\mathcal{O}(n \times \sum_{i=1}^{n} cv(v_i) + n^2 \sum_{i=1}^{n} d(u_i))$, ce qui correspond à $\mathcal{O}(n \times \sum_{i=1}^{n} cv(v_i) + n^3)$.

## Question 5

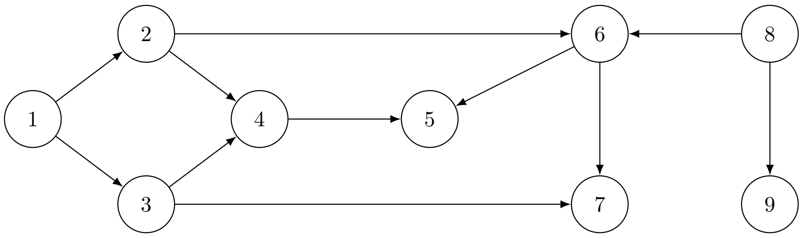
En déduire la complexité en fonction de la représentation.

<u>**Solution**</u>:

- Si $G$ est représenté par une matrice sommet-sommet, $\sum_{i=1}^{n} n = n^2$, on obtient donc du $\mathcal{O}(n^3)$ ;
- Si $G$ est représenté par une matrice sommet-arête, $\sum_{i=1}^{n} mn = mn^2$, on obtient $\mathcal{O}(mn^3)$ ;
- Si $G$ est représenté par des listes d'adjacence, $\sum_{i=1}^{n} d(i) = 2m$ on obtient donc du $\mathcal{O}(n^3 + nm)$.

## Exercice 5 – Parcours génériques d'un graphe orienté

On considère le graphe orienté $G_2 = (V_2, A_2)$ :



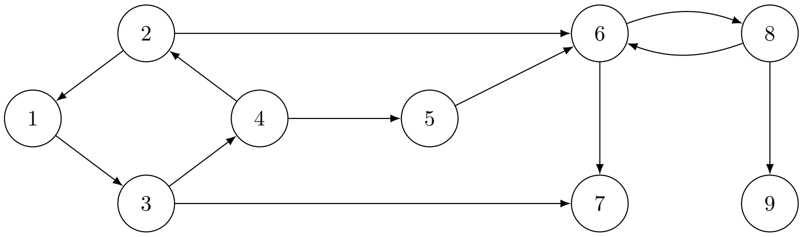
### Question 1

Le graphe $G_2$ admet-il un parcours ? Justifier la réponse.

**Solution**:

Non, car $G_2$ n'a pas de racine.

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



### Question 2

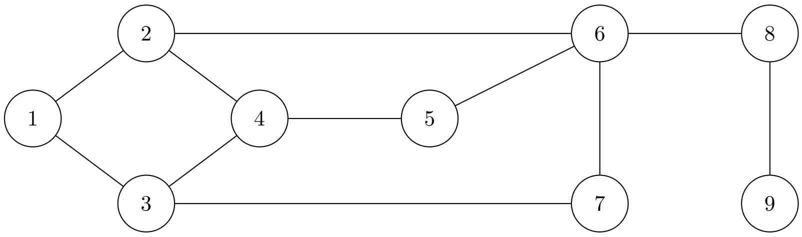
Quelles sont les racines du graphe $G_3$ ? Pour chaque racine, donner un parcours générique de $G_3$.

**Solution**:

Les racines de $G_3$ sont 1, 2, 3 et 4.

Parcours générique de $G_3$ (par exemple) :

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



### Question 3

La liste $(1, 2, 5, 3, 4, 6, 7, 8, 9)$ est-elle un parcours générique de $G_3$ ? Justifier la réponse.

**Solution**:

Non, car le sommet 2 est visité alors qu'il n'est successeur d'aucun sommet déjà visité.

## Exercice 6 – Graphes orientés particuliers

### Question 1

Soit $G = (V, A)$ un graphe orienté ayant $n$ sommets.

1. On suppose que $G$ est composé d'un unique circuit élémentaire. Combien $G$ admet-il de parcours ? Les décrire.
2. On suppose que $G$ est composé d'un unique chemin élémentaire. Combien $G$ admet-il de parcours ? Les décrire.

<u>**Solution**</u>:

1. Soit $(x_1, \ldots, x_n, x_1)$ le circuit élémentaire de $G$, alors $G$ admet $n$ parcours qui sont : $(x_1, \ldots, x_n)$, $(x_2, \ldots, x_n, x_1)$, ..., $(x_n, x_1 \ldots, x_{n-1})$.
2. Soit $(x_1, \ldots, x_n)$ le chemin élémentaire de $G$, alors $G$ admet un seul parcours parcours qui est : $(x_1, \ldots, x_n)$.

## Exercice 7 – Algorithme de calcul d'un parcours d'un graphe orienté

Soit $G = (V, A)$ un graphe orienté.

### Question 1

Comment faut-il définir la bordure d'un sous-parcours $L$ dans le cas d'un graphe orienté pour que l'algorithme de calcul rappelé dans l'exercice 2 reste valable pour un graphe orienté ?

<u>**Solution**</u>:

La bordure d'un sous-parcours $L$ est l'ensemble des sommets de $V$ qui ne sont pas visités par $L$ et qui sont successeurs d'un sommet visité par $L$.

### Question 2

Appliquer l'algorithme du cours au graphe $G_3$ défini dans l'exercice 3 en utilisant la définition de la bordure étendue aux graphes orientés. L'origine à considérer est le sommet 4. Préciser, à chaque itération, le sommet $u$ choisi dans $\mathcal{B}$ (s'il y a plusieurs choix possibles, on prendra le sommet de plus petit numéro), le sous-parcours $L$ et la valeur de la bordure $\mathcal{B}$ de $L$. Les valeurs de $L$ et $\mathcal{B}$ sont celles obtenues à chaque itération en fin du corps de boucle.

<u>**Solution**</u>:

Initialement : $L = (4)$, $\mathcal{B} = \{2, 5\}$
Itération 1 : $u = 2$, $L = (4, 2)$, $\mathcal{B} = \{5, 1, 6\}$
Itération 2 : $u = 1$, $L = (4, 2, 1)$, $\mathcal{B} = \{5, 6, 3\}$
Itération 3 : $u = 3$, $L = (4, 2, 1, 3)$, $\mathcal{B} = \{5, 6, 7\}$
Itération 4 : $u = 4$, $L = (4, 2, 1, 3, 5)$, $\mathcal{B} = \{6, 7\}$
Itération 5 : $u = 5$, $L = (4, 2, 1, 3, 5, 6)$, $\mathcal{B} = \{7, 8\}$
Itération 6 : $u = 6$, $L = (4, 2, 1, 3, 5, 6, 7)$, $\mathcal{B} = \{8, 9\}$
Itération 7 : $u = 8$, $L = (4, 2, 1, 3, 5, 6, 7, 8)$, $\mathcal{B} = \{9\}$
Itération 8 : $u = 9$, $L = (4, 2, 1, 3, 5, 6, 7, 8, 9)$, $\mathcal{B} = \emptyset$

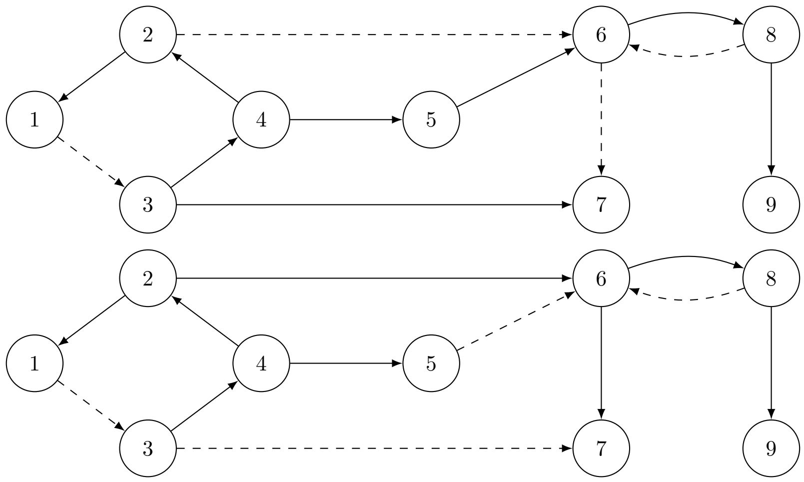
L'algorithme est terminé, il a calculé le parcours $(4, 2, 1, 3, 5, 6, 7, 8, 9)$.

### Question 3

1. Quelle est la définition du graphe de liaison associé à un parcours générique dans le cas d'un graphe orienté $G$ ?
2. Donnez un graphe de liaison pour le parcours $L = (3, 4, 2, 1, 5, 6, 7, 8, 9)$ du graphe $G_3$. Est-il unique ?

<u>**Solution**</u>:

1. $\mathcal{A}(L) = (V, H)$ est un graphe de liaison associé au parcours $L$ si tout sommet $v \in V - \{s\}$ a pour unique prédécesseur un sommet $u \in V$ tel que $u$ est visité avant $v$ dans $L$ et $(u, v) \in A$.
2. Deux graphes de liaisons associés au parcours $L = (3, 4, 2, 1, 5, 6, 7, 8, 9)$ sont représentés à la suite. Les arcs en pointillés ne sont pas des arcs du graphe de liaison.




### Question 4
À faire à la maison. Appliquer l'algorithme au graphe $G_3$ en partant d'un autre sommet ou en suivant une autre stratégie lorsque plusieurs choix sont possibles dans $\mathcal{B}$ (par exemple, prendre le sommet de plus grand numéro, ou choisir le premier sommet qui a été ajouté à $\mathcal{B}$, ou le dernier ajouté, etc.).

## Exercice 8 – Existence d'un parcours d'un graphe orienté
Soit $G = (V, A)$ un graphe orienté.

### Question 1
Soit $G$ un graphe orienté de parcours $L = (s_1, \dots, s_n)$. Montrez qu'il existe un chemin de $s_1$ à $s_i$ pour tout $i \in \{1, \dots, n\}$. Cette démontration peut se faire par récurrence sur $i$.

**Solution**:
Posons $r = s_1$ et montrons, par récurrence forte sur $i$ qu'il existe un chemin de $r$ à $s_i$, pour tout $1 \leq i \leq n$.

**Base :** C'est vrai pour $i = 1$, puisque $r = s_1$.

**Induction :** Soit $1 < i \leq n$ tel qu'il existe un chemin de $r$ à $s_j$, pour tout $1 \leq j < i$. Puisque $L$ est un parcours, il existe $j < i$ tel que $s_i$ est un successeur de $s_j$. Par hypothèse de récurrence, il existe un chemin $\gamma$ de $r$ à $s_j$. La concaténation de $\gamma$ et de l'arc $(s_j, s_i)$ est un chemin de $r$ à $s_i$.

**Conclusion :** il existe un chemin de $s_1$ à $s_i$ pour tout $i \in \{1, \dots, n\}$.

### Question 2
Montrer par l'absurde que, si $r$ est une racine d'un graphe orienté $G$ l'algorithme de construction d'un parcours générique permet de construire un parcours $L$ de $G$.

**Solution**:
Supposons que $r$ est une racine et que l'algorithme ne permet pas de construire un parcours. C'est qu'il existe un sous-parcours $L = (v_1, \dots, v_k)$ avec $k < n$ et tel que $\mathcal{B}(L) = \emptyset$.

Soit alors $v$ un sommet de $V$ qui n'est pas dans $L$. $r$ est une racine, donc il existe un chemin de $r$ à $v$. Comme $v_1 \in L$ et que $v \notin L$, il existe un arc $(u, v) \in A$ avec $u \in L$ et $v \notin L$, et donc $v \in \mathcal{B}(L)$, ce qui est impossible car $\mathcal{B}(L) = \emptyset$.

### Question 3
Démontrez que, pour tout graphe non orienté $G$, $G$ possède un parcours si et seulement si $G$ possède une racine.

**Solution**:
On démontre les deux implications séparément.

$A \Rightarrow B$ **:** Si $G$ possède un parcours, alors, d'après la question 1, $s_1$ est une racine de $G$.

$B \Rightarrow A$ **:** La question 2 a montré que, si $r$ est une racine, on peut toujopurs contruire un parcours générique.