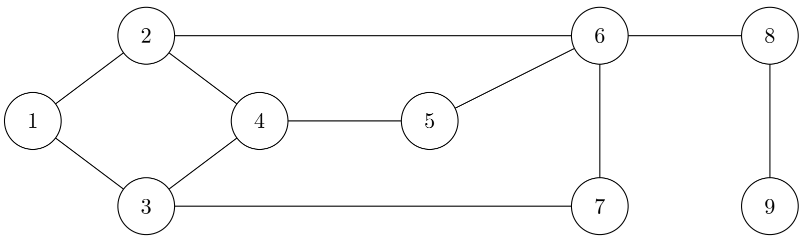
Module LU2IN003 Parcours en profondeur semaine 11

Exercice(s)

Exercice 1 – Graphe non orienté : exercice de base

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

Justifier les réponses négatives.

Question 2
Donner trois parcours en profondeur de G_1, l'un partant du sommet 1, un autre du sommet 9 et un troisième du sommet 5.

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

Exercice 2 – Graphe de liaison en profondeur

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

Question 2
Soit $G = (V, E)$ un graphe non orienté connexe et L un parcours en profondeur de G. Montrer par contradiction que le graphe de liaison en profondeur de L est unique.

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

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

Require: Un graphe non orienté connexe $G = (V, E)$, un sommet s

Ensure: Un parcours en profondeur L d'origine s
  function DFS(G, s)
  $visite[s] := True$, $L := (s)$
  for all arete $\{s, u\} \in E$ **do**
    if not $visite[u]$ **then**
      $L := L$+DFS(G, u)
    end if
  end for
  return L
  end function

$visite$ est un tableau de booléens sur les sommets dont les cases sont initialisées à $Faux$.

Question 1

Exécuter DFS(G_1, 6). Pour cela, donner l'arbre des exécutions et le parcours en profondeur obtenu, ainsi que l'évolution de l'ensemble des sommets visités. Le cas échéant, prendre en priorité le sommet de plus petit numéro.

Question 2

Donnez les valeurs pre et $post$ associées au parcours en profondeur obtenu à la question précédente.

Question 3

1. Que peut-on dire dans le cas général des intervalles $[pre(u), post(u)]$ et $[pre(v), post(v)]$ si (u, v) est un arc de liaison ?
2. Si maintenant $\{u, v\}$ est une arête du graphe qui ne correspond pas à un arc de liaison, mais telle qu'il y a un chemin de u à v dans le graphe de laison en profondeur $\mathcal{A}^(L)$. Que peut-on dire dans le cas général des intervalles $[pre(u), post(u)]$ et $[pre(v), post(v)]$?

Exercice 4 – Complexité du calcul d'un parcours en profondeur

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 profond
eur. On suppose que :

- L est stocké dans une liste circulaire doublement chaînée ;
- $visite$ est un tableau de booléens de taille n initialisé à $Faux$;
- 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. Rappeler sans explication l'ordre de grandeur de $cv(u)$ en fonction de la représentation de G ;

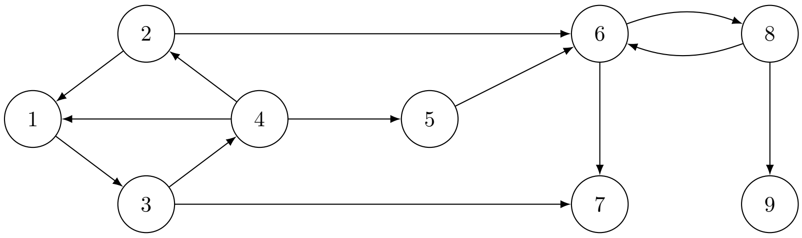
Question 2

1. Quel est l'ordre de grandeur du coût des instructions autres que le calcul des sommets adjacents à un sommet ?
2. Combien de fois chaque sommet est-il traité ?
3. Calculer la complexité de l'algorithme de calcul d'un parcours en profondeur en fonction de cv, puis pour chacune des représentations des graphes non orientés.

Exercice 5 – Graphe orienté : exercice de base

Question 1

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



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

Exercice 6 – Algorithme de calcul d'un parcours en profondeur d'un graphe orienté

Voici un algorithme de calcul d'un parcours en profondeur d'un graphe orienté, dans lequel on calcule aussi les valeurs de pre et $post$:

Require: Un graphe orienté $G = (V, E)$ ayant au moins une racine, une racine s
Ensure: Un parcours en profondeur L d'origine s

```
function DFS(G, s)
  visite[s] := True ; L := (s)
  cpt = cpt + 1 ; pre[s] = cpt
  for all arc (s, u) ∈ E do
    if not visite[u] then
      L := L+DFS(G, u)
    end if
  end for
  cpt = cpt + 1 ; post[s] = cpt
  return L
end function
```

$visite$ est un tableau de booléens sur les sommets dont les cases sont initialisées à $Faux$.
cpt est une variable globale entière, initialisée à 0.
pre et $post$ sont deux tableaux de n entiers, initialisés à 0.

Question 1

Exécuter DFS$(G_1, 4)$ de façon à obtenir le parcours en profondeur $(4, 5, 6, 7, 8, 9, 2, 1, 3)$. Pour cela, donner l'arbre des exécutions et le parcours en profondeur obtenu, ainsi que l'évolution de cpt et celle des tableaux $visite$, pre et $post$ à chaque ouverture et chaque fermeture d'un sommet. On rappelle qu'un sommet u est ouvert à l'appel DFS(G, u) et est fermé au moment où on dépile DFS(G, u).

Question 2

Classifier les arcs de G_3 en fonction du parcours en profondeur $(4, 5, 6, 7, 8, 9, 2, 1, 3)$.

Question 3

On considère le graphe G_3 et le parcours en profondeur $(4, 5, 6, 7, 8, 9, 2, 1, 3)$. Comparer les intervalles $[pre(u), post(u)]$ et $[pre(v), post(v)]$ pour chaque arc avant (u, v), puis pour chaque arc arrière (u, v) puis pour chaque arc transverse (u, v).

Que remarque-t-on ? Pouvez vous l'expliquer ?

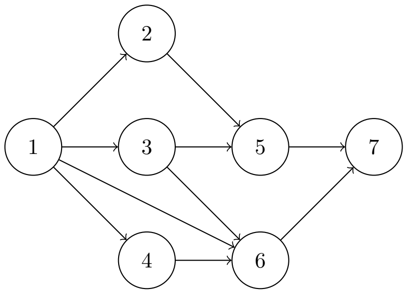
Exercice 7 – Tri topologique et parcours en profondeur

On considère dans cet exercice que $G = (V, A)$ est un graphe orienté sans circuit.

En préambule de cet exercice, vous pouvez regarder la courte vidéo de Christian Laforest :≪ Le parcours en profondeur (DFS) pour planifier (= tri topologique d'un graphe orienté sans circuit) ≫qui explique comment on peut utiliser un parcours en profondeur pour construire un ordre topologique d'un graphe orienté sans circuit.

Question 1

Dans cette question, on considère le graphe orienté sans circuit $G = (V, A)$ représenté par la figure suivante :



1. Exécutez un parcours en profondeur de ce graphe d'origine 1. Vous prendrez en priorité le sommet de valeur minimale quand plusieurs choix sont possibles. Vous préciserez l'arbre des appels, l'ensemble des sommets visités appel par appel, le graphe de liaison et la décomposition des arcs, et les intervalles $[pre(u), post(u)], u \in V$.
2. Est-ce que le graphe possède des arcs arrières pour L ? Pourquoi ?
3. Ordonnez les sommets selon les valeurs $post$ décroissantes. Qu'observez vous ?

Question 2

On suppose dans cette question que $G = (V, A)$ est un graphe orienté quelconque et que L est un parcours en profondeur. Soit alors $(u, v) \in A$ un arc quelconque de G.

— Comparer $post(u)$ et $post(v)$ en fonction du type d'arcs (liaison, avant, arrière, et transverse).
— En déduire que $post(u) < post(v)$ si est seulement si (u, v) est un arc arrière.

Question 3

On suppose maintenant que $G = (V, A)$ est un graphe orienté sans circuit et que L est un parcours en profondeur.

1. Montrez que pour tout arc $e = (u, v) \in A$, $post(u) > post(v)$;
2. Soit alors la liste $p = (v_1, \ldots, v_n)$ composée des éléments de V dans l'ordre des valeurs $post(u)$ décroissante. Montrez que p est un tri topologique de G.

Question 4

En déduire en quelques phrases un algorithme de calcul d'un tri topologique qui utilise un parcours en profondeur.