Algorithmique — Maîtrise
TD 1 : Parcours de graphes

Mihaela Sighireanu, Julien Forest, Yan Jurski

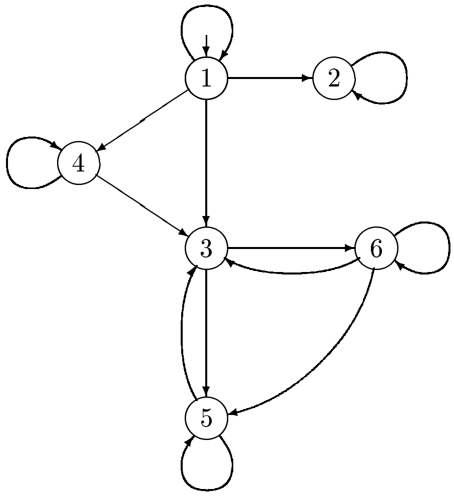

FIG. 1 – Un graphe orienté

Exercice 1 : Donner une représentation du graphe de la figure 1 au moyen d'une liste d'adjacence, puis au moyen d'une matrice d'adjacence.

Exercice 2 : Appliquer à ce graphe l'algorithme de parcours en largeur (le sommet origine est indiqué par une simple flèche entrante). Le résultat sera présenté sur un schéma dans lequel les sommets de profondeur égale seront mis à la même hauteur, le sommet origine étant mis en haut.

Exercice 3 : Appliquer au même graphe l'algorithme de parcours en profondeur. Le résultat sera présenté sous la forme d'une arborescence en profondeur, en distinguant les arcs de liaisons des autres types d'arcs.

Exercice 4 : Donner une version recursive du Parcours en Profondeur. Pourquoi ne vous a t'on pas demandé une version recursive du parcours en largeur.

Exercice 5 : Comparer le temps d'exécution des parcours selon qu'on utilise une représentation par liste d'adjacence ou par matrice d'adjacence.

Exercice 6 : Modifiez l'algorithme de parcours en largeur pour qu'il calcule la profondeur de chaque sommet.

Exercice 7 : Combien y a-t-il de circuits simples dans le graphe de sommets $V = \{v_i, s_i, t_i\}, i \in [0..n-1]$ et d'arcs $E = \{(v_i, s_i), (v_i, t_i), (s_i, v_j), (t_i, v_j)$ avec $j = 1 + i \ mod\ n\}, i \in [0..n-1]$?

Exercice 8 :

1. Quel type de parcours est-il préférable d'utiliser (une seule fois) pour les problèmes suivants :
   (a) trouver un circuit s'il en existe, et renvoyer « faux » sinon ;
   (b) trouver un circuit de longueur minimale contenant un sommet donné, ou renvoyer « faux » s'il n'en existe pas.
2. Le programme suivant est-il une solution de a) ou de b) ? Prouvez-le.

```
Variables Globales:
  booleen marquage[V] initialise a faux
  sommets source[V] initialise a null

Debut :

Variables locales sommets v,x

  afficher ‘‘sommet de depart ?’’
  demander v
  PP(v)
Fin

Procedure PP(sommet v)

Debut
si marquage[v]==vrai return
sinon faire
  marquage[v]=vrai
  pour chaque voisin x de v faire
  si marquage[x]==faux faire
     source[x]=v
     PP(v)
  sinon faire
     afficher ‘‘cycle trouve’’
     afficher x
     tant que (v!=x) faire
          afficher(v)
          v=source[v]
          fin tant que
     s’arreter
  fin sinon
fin sinon

Fin
```

3. Écrivez une solution pour l'autre problème et montrer :
   - qu'elle termine ;
   - que, si on calcule un chemin, celui-ci correspond bien à ce que l'on cherche ;
   - que, si le chemin recherché existe, l'algorithme le trouve.
4. Les trois points précédents sont-ils suffisants pour assurer que les algorithmes sont corrects ?

Exercice 9 : Donner des illustrations pour les définitions du glossaire.

Exercice 10 : On cherche à calculer l'ensemble des sommets atteignables à partir d'un sommet donné dans un graphe $G = (V, E)$. Proposez une solution. Symétriquement on cherche à présent l'ensemble des sommets qui permettent d'atteindre un sommet donné. Alors ?

Glossaire

arc de liaison arc arrivant sur un sommet qui n'a pas encore été découvert dans un parcours en profondeur

arc retour (ou **en arrière**) arc orienté d'un sommet u vers un ancêtre de u dans l'arborescence en profondeur

arc transverse arc « impasse » pour un parcours en profondeur

circuit chemin d'un graphe orienté dont les sommets de départ et d'arrivée sont identiques

cycle (simple) chemin d'un graphe non orienté ne passant jamais deux fois par un même sommet à l'exception des sommets de départ et d'arrivée, qui sont identiques

graphe fortement connexe graphe orienté dans lequel il est possible d'aller de n'importe quel sommet à n'importe quel autre.