Algorithmique Avancée TD01

Hugo Demaret

September 2021

Partie A -

Partie B -

Voir sur :
https://github.com/HugoDemaret/linked_list
https://github.com/HugoDemaret/doublelinked_list
https://github.com/HugoDemaret/binary_tree

Partie C -

Exercice 1 -

Déterminons quelle est la complexité asymptotique de cet algorithme.

Exercice 2 -

Soit T un graphe avec n sommets. Démontrer que les propriétés suivantes sont équivalentes.

(1) T est un arbre
La définition d'un arbre est : Graphe acyclique et connexe.

(2) T est un graphe connexe et acyclique
C'est en fait l'une des définitions d'un arbre.

(3) T est un graphe connexe avec n-1 arrête
Un graphe cyclique à n sommets possède au minimum n arrêtes. Donc un graphe connexe à n-1 arrêtes est acyclique. C'est un arbre.

(4) T est un graphe acyclique avec n-1 arrête

T est un graphe simple. Un graphe acyclique est un graphe simple (la boucle serait un cycle). Le graphe possède n-1 arrêtes, donc chaque sommet possède au moins une arrête (parfois en commun avec un autre sommet). Donc le graphe est connexe. C'est donc un arbre.

Les propriétés précédentes décrivant toutes un arbre, elles sont équivalentes.

Exercice 3 -

Considérons 3 définitions d'un arbre :

Définition 1.
Un arbre T est un graphe :
- Qui est simple (sans boucle)
- Tel que :

$$\forall (u, v) \in T, u \neq v, \exists ! \eta [u, v] \tag{1}$$

Cette définition repose sur les deux propositions suivantes :
Proposition 1 : S'il y a un chemin de v à w dans le graphe G, alors il y a aussi un chemin simple de v à w.
Proposition 2 : S'il y a un cycle dans G, alors il y a un cycle simple dans G.

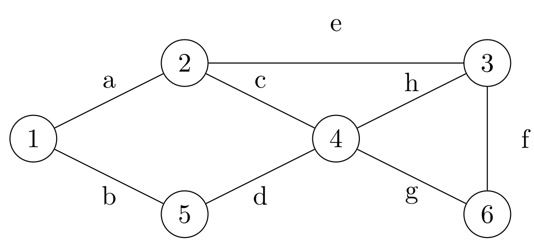
Définition 2.
Un graphe non-orienté connexe et acyclique est appelé un arbre.

Définition 3.
Un arbre est un graphe connexe minimal. C'est-à-dire que toute suppression d'arrêtes le rend non connexe.

Partie D -

Exercice 1.1 -

Soit G le graphe suivant :



Matrice d'adjacence de G :

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

Codage en liste d'arrêtes et sommets :

Sommet	arrêtes	sommets
1	a,b	2,5
2	a,c,e	1,3,4
3	e,f,h	2,4,6
4	c,d,g,h	2,3,5,6
5	b,d	1,4
6	f,g	3,4

Exercice 1.2 -

On considère G un graphe simple et non dirigé.

1 - *Soit A la matrice d'adjacence de G. Que valent les sommes par lignes et par colonnes ?*

Les sommes par lignes et par colonnes nous donnent le degré des sommets de G.

2 - *Soit M la matrice d'incidence de G, c'est-à-dire une matrice de taille $\left|V(G)\right| \cdot \left|E(G)\right|$ telle que $m_{ue} = 1$ si l'arrête e est incidente au sommet u et $m_{ue} = 0$ sinon. Ecrire la matrice d'incidence du graphe de l'exercice précédent.*

Matrice d'incidence de G :

$$M = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

3 - *Que valent les sommes des coefficients par lignes et par colonnes de M ?*

La somme des coefficients par ligne de M nous donne le degré des sommets.
La somme des coefficients par colonne de M nous dit si le graphe est simple ou connexe.

Somme par lignes :

$$\sum \left|M_i\right|$$

Nous donne le degré du sommet i.

$$max(\sum \left|M_i\right|)$$

Nous donne l'ordre du graphe.

Somme par colonnes :

$$\sum \left|M_j\right| =$$

$$\begin{cases} \textit{si 1 : boucle (non-simple) (pas d'information sur la connexité)} \sum \left|M_j\right| = \end{cases}$$

$$\begin{cases} \textit{si 2 : arrête simple et connexe} \end{cases}$$

Nous dit si le sommet est simple, et s'il est connecté au reste du graphe.

Min() somme par colonnes :

$$min(\sum \left|M_j\right|) = \begin{cases} \textit{si 1 : graphe non-simple} \end{cases}$$

$$min(\sum \left|M_j\right|) = \begin{cases} \textit{si 2 : graphe simple et connexe} \end{cases}$$

Nous dit si le graphe est simple et connexe, ou s'il contient une boucle.