TD ALBA - SÉANCE 2
GRAPHES

Exercice 1. Soit $G = (V, E)$ un graphe orienté à n sommets et m arêtes. Soit x un sommet de G, on appelle *degré sortant* de x le nombre d'arêtes qui partent de x et *degré entrant* de x le nombre d'arêtes qui arrivent en x. On note $\deg_s(x)$ le degré sortant de x et $\deg_e(x)$ le degré entrant de x. Montrer que

(1) $\sum_{x\in V} \deg_e(x) = \sum_{x\in V} \deg_s(x)$.

(2) Montrer que $\sum_{x\in V} \deg_e(x) = m$.

Que peut-on dire si G est non orienté sans boucles ?

Exercice 2. Soit G un graphe connexe non-orienté, un *circuit eulérien* de G est un circuit qui passe une et une seule fois par chaque arête de G. On dit que G est un graphe eulérien s'il admet un circuit eulérien.

(1) Montrer que si G est eulérien, alors pour tout sommet x de G, $\deg(x)$ est pair.

(2) Montrer la réciproque.

Exercice 3. Montrer qu'un graphe non-orienté dont tous les sommets ont des degrés supérieurs ou égaux à 2 a un cycle.

Exercice 4. Effectuer un parcours en largeur et en profondeur du graphe non orienté donné par la liste des voisins (i.e. le voisin de 1 est $\{11\}$, les voisins de 2 sont $\{4, 6, 8, 10, 12\}$,…).

$(\{11\}, \{4, 6, 8, 10, 12\}, \{6, 9, 12\}, \{2, 8, 12\}, \{10\}, \{2, 3, 12\}, \{\}, \{2, 4\}, \{3\}, \{2, 5\}, \{1\}, \{2, 3, 4, 6\})$.

Exercice 5 (Correction du parcours en largeur)**.** On considère l'algorithme du parcours en largeur sur un graphe G à partir du sommet s. On note pour tout sommet x de G, $\delta(x)$ sa distance à s (on a $\delta(t) = +\infty$ si t n'est pas dans la composante connexe de s). Montrer par récurrence la propriété $\mathcal{P}(n)$: Il existe un instant dans l'éxécution de l'algorithme tel que

(1) $\delta(x) \le n \Rightarrow d[x] = \delta(x)$.

(2) Si $\delta(x) \le n$ alors un plus court chemin entre x et s est donné par un plus court chemin entre s et $\pi[x]$ complété par l'arête $(\pi[x], x)$.

(3) $\delta(x) > n \Rightarrow d[x] = +\infty$.

(4) $\delta(x) = n$ si et seulement si x est dans la file F, que l'on note F_n à cet instant.

Exercice 6. Soit G un graphe non orienté à n sommets ($n \ge 1$).

(1) Montrer que si G est connexe, alors il possède au moins $n - 1$ arêtes.

(2) On dit que G est un *arbre* s'il est connexe et sans cycle. Montrer les équivalences suivantes.

  (a) G est un arbre.

  (b) G ne contient pas de cycle et possède $n - 1$ arêtes.

  (c) G est connexe et possède $n - 1$ arêtes.

  (d) G est connexe et pour tout arête e de G le graphe $G \setminus e$ n'est plus connexe.

(e) Pour tout sommet u, v de G il existe un unique chemin allant de u à v.

(f) G est sans cycle mais l'addition d'une arête crée exactement un cycle.

Exercice 7. On considère en C le code suivant :

```
typedef struct Pile Pile;
struct Pile
{
    int head;
    Pile* next;
};


typedef struct File File;
struct File
{
    int head;
    File* next;
};
```

Donner pour la structure de Pile et de File le code de la fonction *pop* et *add*.

Exercice 8. Écrire l'algorithme de parcours en profondeur en utilisant une pile.