Exercice 1

Soit $f(n) = \frac{1}{3}n\left(n + \frac{1}{2}\right)(n+1)$. Montrer que $f(n) = \mathcal{O}(n^3)$

Les relations suivantes sont elles vraies ?

i. $f(n) = \mathcal{O}(n^{10})$
ii. $f(n) = \frac{1}{3}n^3 + \mathcal{O}(n^2)$
iii. $\mathcal{O}(n^2) = \mathcal{O}(n^3)$
iv. $\mathcal{O}(n^3) = \mathcal{O}(n^2)$
v. $\frac{1}{3}n^3 + \mathcal{O}(n^2) = \mathcal{O}(n^3)$

Exercice 2

Dessiner le graphe G_1 non orienté décrit par les listes d'adjacence ci-dessous. Donner la matrice d'adjacence et la matrice d'incidence de G_1 (on numérotera les arêtes suivant l'ordre lexicographique sur le couple de sommets *extrémités*).

```
1 : 2, 3
2 : 1, 3, 4, 5
3 : 1, 2, 5
4 : 2, 5
5 : 2, 3, 4
```

Exercice 3

Donner les listes d'adjacence, la matrice d'adjacence et la matrice d'incidence du graphe orienté de la FIG. 1.

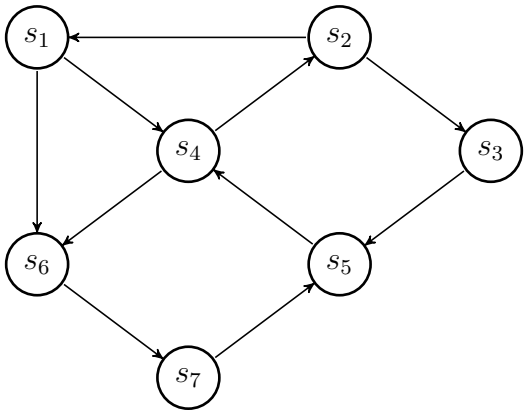

FIGURE 1 – G_2

Exercice 4

Appliquer l'algorithme du ***parcours en largeur*** `PL(G, s)` au graphe G_1 (Exercice 2) à partir du sommet `1` et au graphe G_2 (Exercice 3) à partir du sommet s_1. Chaque fois donner l'ordre d'entrée des sommets dans la file.

```
0   PL(G,s)
1     pour chaque sommet u de X[G] \ {s} faire
2         couleur[u] <- BLANC
3         d[u] <- infini
4         pere[u] <- nil
5     couleur[s] <- GRIS
6     d[s] <- 0
7     pere[s] <- nil
8     Enfiler(F, s)
9     tant que non vide(F) faire
10        u <- tete(F)
11        pour chaque v de Adj(u) faire
12            si couleur[v] = BLANC
13                alors couleur[v] <- GRIS
14                      d[v] <- d[u] + 1
15                      pere[v] <- u
16                      Enfiler(F, v)
17        Defiler(F)
18        couleur[u] <- NOIR
```

Exercice 5

On se donne un graphe orienté à n sommets et m arcs. On suppose que le degré (nombre d'arcs entrant ou sortant) de chaque sommet est au plus d.

Donner l'algorithme et une majoration (la plus basse possible en fonction de n, m, d) pour la complexité de chacune des opérations suivantes :

1. faire afficher tous les successeurs d'un sommet s donné
2. faire afficher tous les prédécesseurs d'un sommet s donné
3. déterminer si un graphe est sans boucle (une boucle est un arc $s \to s$)
4. effectuer le parcours en largeur à partir d'un sommet donné

suivant qu'on utilise, pour représenter le graphe, une matrice d'adjacence `A[i,j]`, une matrice d'incidence `B[u,e]`, ou des listes de successeurs `Adj[u]`.

Exercice 6

On considère un graphe orienté à n sommets et m arcs. On dit que deux arcs e_1 et e_2 sont opposés si $e_1 = (i, j)$ et $e_2 = (j, i)$ avec i et j deux sommets distincts du graphe.

Pour chacune des trois représentations (listes de successeurs, matrice d'adjacence, matrice d'incidence), écrire un algorithme pour déterminer si le graphe contient des arcs opposés et en étudier la complexité.