TP n°23 – Manipulation des graphes

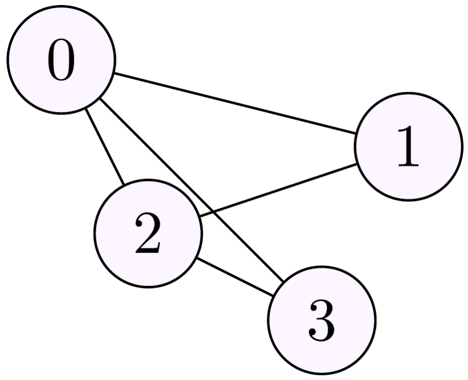
1 Manipulations de graphes

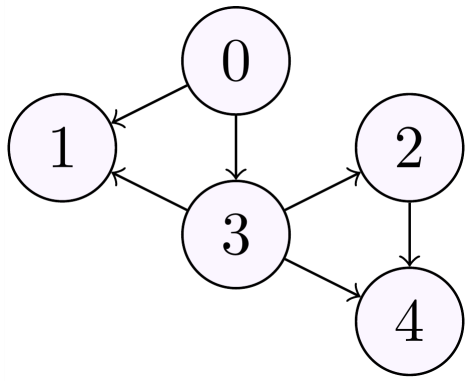
On rappelle qu'il existe deux représentations classiques pour un graphe $G = (S, A)$ avec $n = |S|$ sommets :

- la matrice d'adjacence : matrice $n \times n$, remplie de 0 et de 1, un 1 dans la case (i, j) indiquant la présence d'une arrête entre i et j. (une arc de i à j pour le cas orienté)
- la liste d'adjacence : tableau à n cases dont chaque case i contient la liste des sommets accessibles depuis le sommet i.

  Dans le cas de C, on a pas de type liste donc on utilisera un tableau de tableaux et chaque tableau interne aura comme premier élément sa taille.

- **Q1.** Définir en C la matrice d'adjacence et la liste d'adjacence des graphes suivants :





- **Q2.** Écrire une fonction en C qui compte le nombre d'arêtes à partir de la matrice d'adjacence d'un graphe non-orienté. Avez-vous besoin d'une autre entrée ?
- **Q3.** Faire pareil mais avec la représentation par liste d'adjacence.
- **Q4.** Écrire une fonction en C qui étant donné un graphe non orienté sous forme de matrice d'adjacence renvoie le tableau des degrés des sommets.
- **Q5.** Faire pareil mais avec la représentation par liste d'adjacence.
- **Q6.** Écrire une fonction permettant d'obtenir le tableau des successeurs d'un sommet i pour un graphe orienté représenté par une liste d'adjacence.
- **Q7.** Faire la même chose mais avec les prédecesseurs d'un sommet i. Il est préférable de calculer leur nombre avant d'essayer de les stocker.

On appelle graphe complémentaire de $G = (S, A)$ le graphe $\overline{G} = (S, (S \times S) \setminus A)$.

On appelle graphe inverse (ou transposé) de $G = (S, A)$, orienté, le graphe G^T obtenu en considérant les mêmes sommets et toutes les arcs de G, mais inversés. ((x, y) devient (y, x))

À partir d'ici on va passer en Ocaml. Les matrices et listes d'adjacence des deux graphes de la question 1 sont définis dans le fichier tp26.ml

- **Q8.** Écrire une fonction en Ocaml qui calcule le graphe complémentaire d'un graphe représenté sous forme de matrice d'adjacence.
- **Q9.** Faire pareil avec une représentation sous forme de listes d'adjacence.
- **Q10.** Écrire une fonction en Ocaml qui calcule le graphe inverse d'un graphe représenté sous forme de matrice d'adjacence.
- **Q11.** Faire pareil avec une représentation sous forme de listes d'adjacence.
- **Q12.** Écrire en Ocaml une fonction qui calcule la matrice d'adjacence du sous-graphe induit d'un graphe $G = (S, A)$ par un sous-ensemble $W \subset S$ de sommets. Si $|W| = m$, on attend en sortie une matrice de taille $m \times m$. En machine, W sera représenté par une liste.

  On rappelle que la représentation par matrice d'adjacence (ou liste d'adjacence) n'a de sens que si les sommets sont numérotés à partir de 0. On pourra donc rénuméroter les sommets de W de 0 à m-1.
- **Q13.** Faire pareil en prenant en entrée une liste d'adjacence et en renvoyant la liste d'adjacence du sous-graphe induit

2 Coloriage

Un coloriage de graphe consiste à associer à chaque sommet du graphe une couleur sans que deux sommets côte à côte n'aient la même couleur.

Un graphe est k-coloriable s'il est possible de colorier le graphe avec k couleurs au plus.

Informatiquement, on peut représenter les couleurs des entiers. Les couleurs des sommets peuvent être représentées par un tableau, chaque case i contenant la couleur du sommet i.

- **Q14.** Imaginer, sans le programmer, un algorithme glouton qui permet de déterminer si un graphe est k-coloriable.
- **Q15.** Quelle est sa complexité ?
- **Q16.** Quelle représentation est la plus adaptée ? La matrice ou les listes ?
- **Q17.** Programmer l'algorithme dans le langage de votre choix.