

---

# TP n° 3: Manipulation des tableaux en Java

---

## Objectif du TP

Ce TP a pour objectif de présenter à l'étudiant la possibilité de représenter des problèmes réels par des structures de tableaux. À la fin de ce TP, il sera en mesure de détecter les limitations des tableaux et de traduire un algorithme en un programme Java.

## 1 Modélisation des graphes

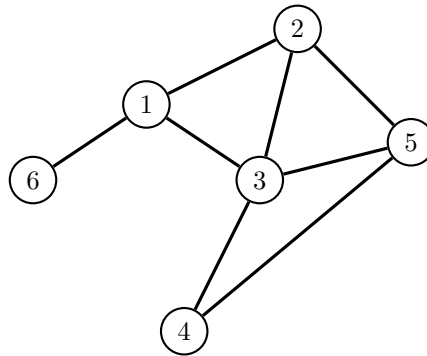
### Première partie

Un graphe  $G$  est un couple  $G = (S, A)$  où :

- $S$  est un ensemble fini de sommets (ou de noeuds)
- $A$  est un ensemble fini d'arêtes (ou d'arcs)

Dans l'exemple ci-dessous, le graphe  $G$  est composé de l'ensemble des sommets  $S = \{1, 2, 3, 4, 5, 6\}$  et de l'ensemble d'arêtes  $A = \{(1, 2), (1, 3), (1, 6), (2, 3), (2, 5), (3, 4), (3, 5), (4, 5)\}$

On appelle  $G$  un graphe **non orienté** (les arcs n'ont aucune orientation) et **non valué** (aucune valeur n'est attribuée aux arcs)



1. Écrire un programme en Java qui modélise un graphe non orienté et non valué en utilisant une structure de tableau (à une dimension ou à 2 dimensions). Le nombre de sommets et les arêtes sont saisis par l'utilisateur.

**Remarque :** Saisir les arêtes équivalentes qu'une seule fois (par exemple  $(1, 2)$  et  $(2, 1)$  sont équivalentes)

2. Écrire une méthode qui affiche les sommets voisins d'un sommet donné.

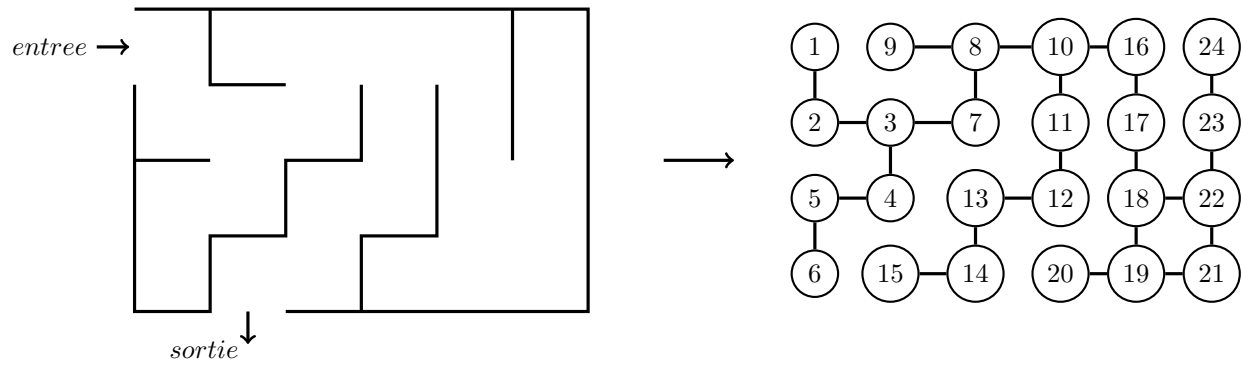
### Deuxième partie

Les algorithmes de **parcours** font partie des algorithmes utilisés sur les graphes non orientés en algorithmique. Ils permettent de parcourir le graphe sans boucler indéfiniment dans les circuits fermés.

L'une des applications les plus répandues des algorithmes de parcours des graphes non orientés est **La sortie du labyrinthe**. Pour pouvoir sortir d'un labyrinthe il faut trouver le seul chemin qui mène à la sortie sans tomber dans les circuits fermés. En algorithmique, on peut modéliser ce processus comme suit :

1. Si la case courante  $C$  est la sortie alors **Terminé**
2. Sinon **marquer**  $C$ .
3. Pour chaque case  $V$  voisine de  $C$  faire
  - (a) Si  $V$  n'est pas marqué alors  $C \leftarrow V$  et revenir à 1

Les labyrinthes sont généralement modélisés par des graphes non orientés et non valués comme le montre la figure suivante :



Écrire un programme Java qui résout le problème de sortie du labyrinthe en implémentant l'algorithme mentionné sur le graphe ci-dessous.

## 2 Calculatrice binaire

1. En utilisant les tableaux, écrire un programme Java qui convertit un nombre entier et/ou réel en binaire. Ne pas utiliser les méthodes prédéfinies de Java.
2. Utiliser les tableaux pour additionner, soustraire, multiplier et diviser deux nombres binaires sans les convertir en décimal. Les opérations doivent être définies dans des méthodes.
3. Étendre le programme pour calculer plusieurs opérations en une seule ligne avec un ordre de priorité de gauche à droite.