

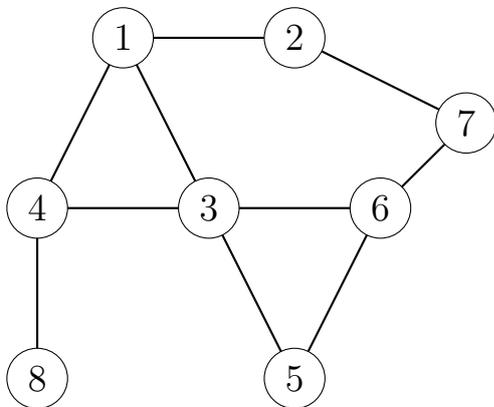
## Algorithmique (AL5)

### TD n° 2 : parcours en largeur de graphes non orientés

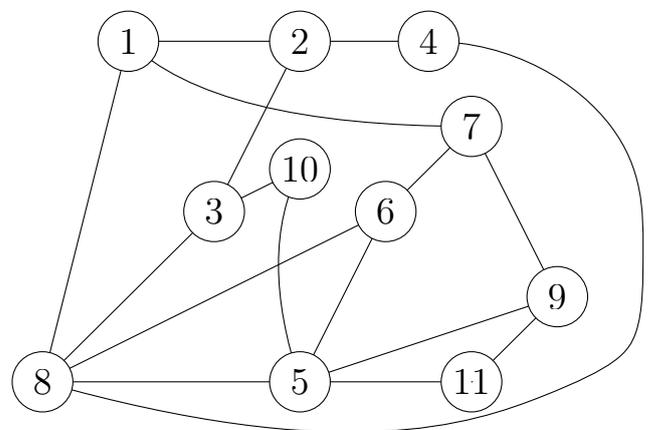
Les exercices notés avec une étoile \* présentent un niveau de difficulté plus élevé.

#### Exercice 1 : parcours en largeur

Appliquer aux deux graphes ci-dessous l'algorithme de parcours en largeur à partir du sommet 1. Pour chaque graphe, donner l'arbre résultant de ce parcours.



Un graphe



un autre graphe

#### Exercice 2 : parcours et matrice d'adjacences

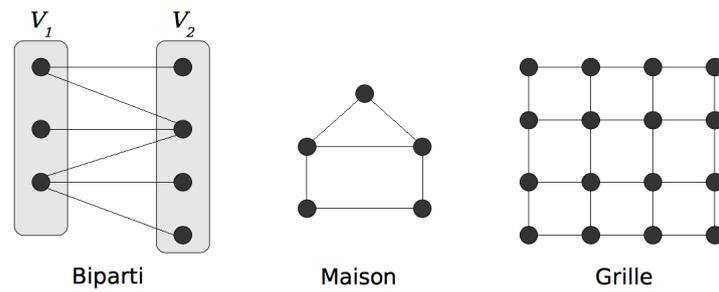
Quelle est la complexité de l'algorithme de parcours en largeur d'un graphe non orienté si celui-ci est représenté par une matrice d'adjacences ?

#### Exercice 3 : plus courts chemins

Proposer un algorithme de complexité linéaire en  $(n + m)$  (nombre de sommets plus nombre d'arêtes) qui calcule le plus court chemin dans un graphe non orienté entre deux sommets donnés.

#### Exercice 4 : composantes connexes

Proposer un algorithme de complexité linéaire en  $(n + m)$  (nombre de sommets plus nombre d'arêtes) qui calcule les composantes connexes d'un graphe non orienté.

FIGURE 1 – Schéma général d'un graphe biparti, la maison, la grille  $G_{4,4}$ **Exercice 5 : Graphes bipartis**

Un graphe  $G = (V, E)$  est un graphe biparti si l'ensemble des sommets  $V$  peut être partitionné en deux sous-ensembles  $V_1$  et  $V_2$  (i.e.  $V_1 \cap V_2 = \emptyset$  et  $V_1 \cup V_2 = V$ ), de sorte que les arêtes de  $E$  ont exactement une extrémité dans  $V_1$  et une extrémité dans  $V_2$ .

1. Indiquer si la maison est un graphe biparti.
2. Faire de même pour la grille  $G_{4,4}$ .
3. Montrer qu'un graphe est biparti ssi il n'a pas de cycle de longueur impaire. La longueur d'un cycle est le nombre d'arêtes qui le composent. (Astuce : on peut se servir d'une 2-coloration).

**Remarque :** Colorier un graphe revient à mettre des couleurs sur chaque sommet du graphe. La seule contrainte imposée est que deux sommets adjacents ne soient pas de la même couleur.

4. Proposer un algorithme qui prend en entrée un graphe, qui teste si le graphe est biparti, et qui renvoie un cycle impair sinon. On pourra se servir du parcours en largeur (BFS) vu en cours.
5. Évaluer la complexité de votre algorithme.

**Exercice 6 : cycle\***

Proposer un algorithme renvoyant, s'il existe, un cycle de longueur minimale passant par un sommet  $s$  donné dans un graphe  $G$  non orienté. Prouver sa correction.