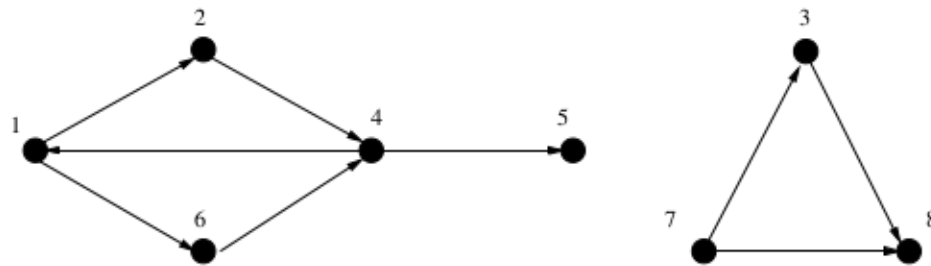


TD2 : parcours

1 Parcours en largeur et applications

1. Appliquer l'algorithme de parcours en largeur pour le graphe ci-dessous.
2. **Composantes connexes.**
 - (a) En utilisant le parcours en largeur, calculez le nombre de composantes connexes d'un graphe non orienté.
 - (b) Plus précisément montrer comment modifier l'algorithme de parcours en largeur pour qu'il étiquette chaque sommet avec le numéro de sa composante connexe.
 - (c) Modifier l'algorithme précédent afin d'obtenir toutes les composantes connexes sous la forme d'une liste de listes de sommets.



3. **Recherche d'un plus court chemin.** Soit $G = (X, A)$ un graphe (orienté ou non-orienté). Soient x et y deux sommets. La **distance** $\delta(x, y)$ entre les sommets x et y est la longueur d'un plus court chemin entre x et y si un tel chemin existe. Dans le cas contraire, la distance sera par définition $+\infty$.
 - (a) Etant donné un sommet $s \in X$, écrire un algorithme qui calcule $\delta(s, x)$ pour tout sommet x du graphe.
 - (b) Proposer un algorithme qui permet de déterminer un plus court chemin, et donc la distance, entre deux sommets fixés d'un graphe (orienté ou pas).

2 Parcours en profondeur

1. Répondre aux deux premières parties de l'exercice précédent en remplaçant le parcours en largeur par le parcours en profondeur.
2. Proposer un algorithme itératif du parcours en profondeur.