

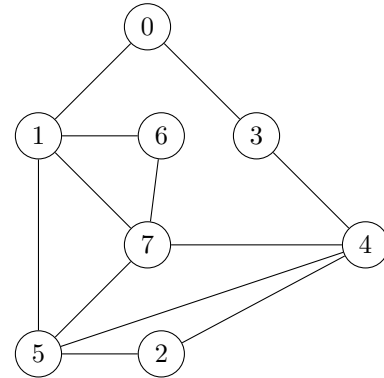
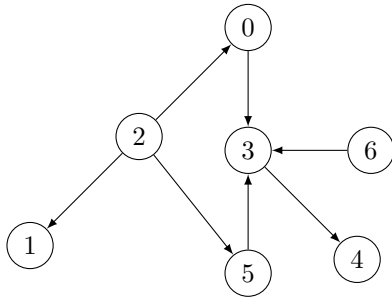
TP 11 - Graphes

Dans ce TP, on aborde l'implémentation des graphes par listes d'adjacence et matrice d'adjacence sur des questions simples. Dans tout ce TP les sommets des graphes sont supposés être les entiers de 0 à $n - 1$ où $n = |S|$.

1 Implémentation creuse : listes d'adjacence

Dans cette partie, les graphes sont représentés par listes d'adjacence.

Question 1. Définir des variables représentant les graphes suivants :



Question 2. Ecrire une fonction `sansBoucleLA` qui, étant donné un graphe, renvoie un booléen indiquant si ce graphe ne contient aucune boucle.

Indiquer la complexité de l'algorithme en fonction de $|S|$ et $|A|$.

Question 3. Ecrire une fonction `degreSortantLA` qui, étant donné un graphe, renvoie une liste telle que la case d'indice i contienne le degré sortant du sommet i .

Indiquer la complexité de l'algorithme en fonction de $|S|$ et $|A|$.

Question 4. Ecrire une fonction `degreEntrantLA` qui, étant donné un graphe, renvoie une liste telle que la case d'indice i contienne le degré entrant du sommet i .

Indiquer la complexité de l'algorithme en fonction de $|S|$ et $|A|$.

Question 5. Ecrire une fonction `VersOrienteLA` qui, étant donné un graphe non orienté, renvoie le graphe obtenu après orientation des arêtes dans l'ordre croissant des sommets, c'est-à-dire pour tout arête $\{s, t\}$ du graphe de départ, on crée l'arc (s, t) .

Question 6. Ecrire une fonction `VersNonOrienteLA` qui, étant donné un graphe orienté, renvoie le graphe obtenu en supprimant l'orientation des arcs.

Une **clique** d'un graphe non orienté G est un ensemble de sommets X tel que pour tout $x, y \in X$ avec $x \neq y$, il y a une arête entre x et y i.e. $\{x, y\} \in A$.

Question 7. Ecrire une fonction `estCliqueLA` qui, étant donné un graphe non orienté et une liste de sommets, renvoie un booléen indiquant si cet ensemble de sommets est une clique pour le graphe considéré.

Indiquer la complexité de l'algorithme en fonction de $|S|$ et $|A|$.

Colorier un graphe, c'est donner une couleur à chacun de ses sommets en faisant en sorte que deux sommets voisins n'aient jamais la même couleur. En pratique un tel coloriage est donné sous la forme d'une liste telle que la case d'indice i contient l'entier correspondant à la couleur du sommet i .

Question 8. Ecrire une fonction `bienColorierLA` qui, étant donné un graphe et une liste représentant un coloriage de ce graphe, renvoie un booléen indiquant si ce coloriage respecte les contraintes exposées ci-dessus.

Indiquer la complexité de l'algorithme en fonction de $|S|$ et $|A|$.

2 Implémentation dense : matrice d'adjacence

Dans cette partie, les graphes sont représentés par matrice d'adjacence.

Question 9. Définir des variables représentant les graphes suivants de la question 1.

Question 10. Ecrire une fonction `sansBoucleMA` qui, étant donné un graphe, renvoie un booléen indiquant si ce graphe ne contient aucune boucle.

Indiquer la complexité de l'algorithme en fonction de $|S|$ et $|A|$. Comparer avec la complexité obtenue dans le cas de l'implémentation creuse.

Question 11. Ecrire une fonction `degreSortantMA` qui, étant donné un graphe, renvoie une liste telle que la case d'indice i contienne le degré sortant du sommet i .

Indiquer la complexité de l'algorithme en fonction de $|S|$ et $|A|$. Comparer avec la complexité obtenue dans le cas de l'implémentation creuse.

Question 12. Ecrire une fonction `degreEntrantMA` qui, étant donné un graphe, renvoie une liste telle que la case d'indice i contienne le degré entrant du sommet i .

Indiquer la complexité de l'algorithme en fonction de $|S|$ et $|A|$. Comparer avec la complexité obtenue dans le cas de l'implémentation creuse.

Question 13. Ecrire une fonction `VersOrienteMA` qui, étant donné un graphe non orienté, renvoie le graphe obtenu après orientation des arêtes dans l'ordre croissant des sommets, c'est-à-dire pour tout arête $\{s, t\}$ du graphe de départ, on crée l'arc (s, t) .

Question 14. Ecrire une fonction `VersNonOrienteMA` qui, étant donné un graphe orienté, renvoie le graphe obtenu en supprimant l'orientation des arcs.

Question 15. Ecrire une fonction `estCliqueMA` qui, étant donné un graphe non orienté et une liste de sommets, renvoie un booléen indiquant si cet ensemble de sommets est une clique pour le graphe considéré.

Indiquer la complexité de l'algorithme en fonction de $|S|$ et $|A|$. Comparer avec la complexité obtenue dans le cas de l'implémentation creuse.

Question 16. Ecrire une fonction `bienColorierMA` qui, étant donné un graphe et une liste représentant un coloriage de ce graphe, renvoie un booléen indiquant si ce coloriage respecte les contraintes exposées ci-dessus.

Indiquer la complexité de l'algorithme en fonction de $|S|$ et $|A|$. Comparer avec la complexité obtenue dans le cas de l'implémentation creuse.

3 D'une implémentation à l'autre

Si nécessaire, il faut être capable de passer d'une représentation creuse à une représentation dense et inversement.

Question 17. Ecrire une fonction `creuseVersDense` qui prend en entrée un graphe représenté sous forme de listes d'adjacence et renvoie la représentation de ce même graphe sous forme de matrice d'adjacence.

Question 18. Ecrire une fonction `DenseVersCreuse` qui prend en entrée un graphe représenté sous forme de matrice d'adjacence et renvoie la représentation de ce même graphe sous forme de listes d'adjacence.