

Initiation à l'informatique - Premier semestre : TP 4

Type abstrait centré-arêtes

Dans ce TP, on utilisera uniquement le type abstrait graphe appelé *type abstrait centré-arêtes*.

Exercice 4.1

Écrire une fonction, `exist_edge (G, s, t)`, qui renvoie 1 si il existe une arête entre deux sommets donnés, 0 sinon.

En utilisant cette fonction, écrire une fonction, `is_path (G, u, v, w)`, qui renvoie 1 si il existe une chaîne simple contenant uniquement les trois sommets `u`, `v` et `w`, sinon 0.

Exercice 4.2

Écrire une fonction `display_neighbors (G, s)`, qui affiche tous les voisins d'un sommet `s` donné.

En modifiant cette fonction, écrire une fonction `neighbor (G, s, i)` permettant d'obtenir le i^{eme} sommet voisin d'un sommet `s` (si n est le nombre de voisins de `s`, i est un entier compris entre 0 et $n - 1$). Quelle est la complexité de votre fonction?

Exercice 4.3

Écrire une fonction permettant de déterminer si un graphe possède des arêtes multiples. Quelle est la complexité de votre fonction? (Astuce : écrivez une fonction `compare (e1, e2)` qui retourne 1 si les deux arêtes ont les mêmes extrémités, 0 sinon)

Exercice 4.4

Écrire une fonction `unmark_all_nodes (G)` qui démarque tous les sommets du graphe `G`.

En utilisant les fonctions `UNMARK_NODE` et `MARK_NODE`, écrire une fonction qui détermine s'il existe un sommet isolé dans un graphe. Quelle est la complexité de vos fonctions?