

TP n°7 - Graphes : parcours

Notions abordées

- Notion de bordure d'un ensemble de sommets
- Parcours d'un graphe, parcours en largeur, parcours en profondeur
- Plus court chemin en nombre d'arcs
- Décomposition en composantes connexes d'un graphe non orienté
- Utilisation du modules `collections.deque` pour implémenter des piles de des files.

 Toutes les fonctions doivent être commentées et testées. Elles doivent notamment être munies d'une description faisant suite à d'éventuelles hypothèses sur leurs arguments placée entre triple guillemets sous la signature de la fonction, (`""" documentation sur plusieurs lignes """`). Les autres commentaires s'écrivent sur des lignes commençant par dièse : `# ligne de commentaire`.

Exercice 0 Piles et files en Python

Question 1

Rappeler les opérations des types abstraits `pile` et `file`.

Question 2

En utilisant une liste Python pour enregistrer les données comment serait implémenté le type abstrait `pile`. Expliquer et préciser les complexités de chacune des opérations.

Question 3

Même question pour le type abstrait `file`.

Le module `collections.deque` permet de gérer des structures de données semblables aux listes dans lesquelles les opérations d'ajout et de suppression sont en $O(1)$, à la fois en tête et en fin (on peut repenser aux listes doublement chaînées vues en C).

Ces opérations sont implémentées par les méthodes suivantes :

- `append(elem)` ajoute `elem` à la fin (*i.e.* à droite)
- `appendleft(elem)` ajoute `elem` au début (*i.e.* à gauche)
- `pop()` supprime le dernier élément *i.e.* le plus à droite (pourvu que la structure soit non vide)
- `popleft()` supprime le premier élément *i.e.* le plus à gauche (pourvu que la structure soit non vide)

La documentation complète de ce module est disponible à l'adresse suivante :

<https://docs.python.org/fr/3/library/collections.html#collections.deque>

Question 4

Dans un interpréteur créer une `deque` vide, puis y insérer par la gauche les entiers 3,4,5,6. Quelles

opérations permettent de remplacer 3 par 2 ? Les réaliser. Supprimer le dernier éléments en fin et l'ajouter en tête.

Question 5

En utilisant une *deque* pour enregistrer les données, comment seraient implémentés les types abstraits *pile* et *file* ? Expliquer et préciser les complexités de chacune des opérations.

Exercice 1 Parcours en largeur

Question 1

Quelle représentation d'un graphe non orienté vous paraît la plus adaptée pour réaliser un parcours en largeur de ce graphe ? Justifier pas des arguments de complexité.

Question 2

Définir une fonction `parcours_larg` qui prend en entrée un graphe non orienté (sous une forme à préciser) et qui renvoie un parcours an largeur de ce graphe. Préciser la complexité de cette fonction.

Si u et v sont deux sommets d'un graphe non orienté $G = (V, E)$, la distance (en nombre d'arêtes) entre u et v , notée $d(u, v)$, vaut $+\infty$ si u et v ne sont reliés par aucune chaîne dans G , et la longueur minimum d'une chaîne entre u et v dans G sinon.

Question 3

Définir une fonction qui prend en entrée un graphe non orienté (sous une forme à préciser) et un sommet s , et qui renvoie une liste indiquant pour chaque sommet sa distance à s (en nombre d'arêtes).

Question 4

Définir une fonction qui prend en entrée un graphe non orienté (sous une forme à préciser) et deux sommets u et v , et qui la distance entre u et v (en nombre d'arêtes).