

Initiation à l'Informatique

Durée : 2h ; tous documents autorisés

Vous utiliserez le type abstrait graphe centré-arêtes.

- Lancez Emacs et ouvrez un fichier `TP.py`. Ce fichier contiendra les fonctions à implémenter. Pensez à sauvegarder votre travail!
- Démarrez l'interpréteur Python.
- Importez le module `egraph`.
- Chargez le fichier `graph2.py`. Ce fichier contient la définition d'un graphe `gr2`.

Exercice 1 *Découverte du graphe*

Nous noterons s_i le sommet numéro i et e_j l'arête numéro j de `gr2`. Répondez sur la feuille aux questions suivantes :

- Quel est le nombre d'arêtes de `gr2` ?

Quelle opération utilisez-vous ?

- Quel est le nombre de sommets de `gr2` ?

Quelle opération utilisez-vous ?

- Quelles sont les extrémités de e_0 ?

Quelle opération utilisez-vous ?

- Quelles sont les extrémités de chacune des autres arêtes ?

- Dessinez le graphe `gr2`.

Exercice 2 *Degré d'un sommet*

Écrivez (dans le fichier `TP.py`) une fonction `degre(G,s)` qui renvoie le degré du sommet `s` dans le graphe `G`.

Testez cette fonction sur le graphe `gr2` et les sommets s_0 , puis s_1 .

- Expliquez comment vous effectuez ces tests.
- Quels résultats obtenez-vous ?

Exercice 3 *Graphes eulériens*

On considère un graphe G connexe. Un graphe connexe admet une chaîne eulérienne si et seulement si le nombre de sommets de degré impair vaut 0 ou 2.

Écrivez une fonction `eulerien(G)` qui renvoie 1 si le graphe connexe G est eulérien et 0 sinon.

Testez cette fonction sur le graphe `gr2`.

- Quel résultat obtenez-vous ?
- Faites un dessin illustrant ce résultat.

Exercice 4 *Marquer les voisins*

Écrivez une fonction `marquer_voisins(G,s)` qui marque les sommets voisins du sommet s dans le graphe G .

Exercice 5 *Fédérateurs*

On dit qu'un ensemble de sommets $\{t_0, \dots, t_k\}$ d'un graphe G est fédérateur si tout sommet s de G est voisin de l'un des sommets t_i .

On fixe ici $k = 2$, et on cherche s'il existe un couple de sommets (s_1, s_2) fédérateur du graphe G .

Un algorithme peut être le suivant :

1. Décrire les couples de sommets (s_1, s_2) .
2. Pour chaque couple, démarquer tous les sommets de G , puis
3. Marquer les voisins de s_1 et s_2 .
4. Regarder si tous les sommets sont marqués : si oui, afficher s_1 et s_2 ; sinon, passer au couple suivant et repartir en 2.
5. Si tous les couples ont été essayés sans succès, afficher : "Pas de couple fédérateur"

À l'aide de cet algorithme,

- Écrivez une fonction `federateur(G)` qui affiche un couple fédérateur du graphe G si existe et un message explicatif sinon.
- Testez cette fonction sur le graphe `gr2`.

Copiez votre fichier `TP.py` :
`cp TP.py ~jcaval/TPII/G /TP.py`

GROUPE

NOMS :