

TP3 Algo Avancée

Introduction :

Dans le cadre de ce TP, nous allons explorer les graphes. Dans ce TP, un graphe est représenté par une liste chaînée de sommets. Le type sommet est une structure qui contient plusieurs champs dont l'un d'eux est un pointeur vers le sommet suivant.

Nos algorithmes :

a. Parcours en largeur :

Dans notre algorithme de parcours en largeur, nous vérifions d'abord si le graphe passé en paramètre est vide ou pas.

Une fois la vérification faite, nous créons une file où on enfile le sommet de départ en premier.

Ensuite nous entrons dans notre boucle principale, tant que la file est non vide, nous entrons dans une nouvelle itération.

Dans chaque itération, nous défilons la file (On prend le premier élément qui a été ajouté à la file tout en le supprimant de la file), puis si le sommet récupérer est non visité, nous le visitons : Nous affichons son label puis on parcourt sa liste d'arcs sortant afin d'enfiler les sommet destination pas encore visités.

b. Parcours en profondeur :

L'algorithme d'affichage en profondeur a pour but d'aller le plus loin dans un chemin, puis quand tous les sommets destinations ont déjà été visités, on revient en arrière.

Dans notre implémentation, nous adopterons une approche récursive. Nous vérifions d'abord que le graphe n'est pas vide. Ensuite, nous parcourons la liste des arcs du sommet du graphe passé en paramètre.

Pour chaque sommet destination non visité, on passe le champ visite à 1 pour indiquer que ce sommet a été traité, avant d'afficher son label et d'appeler la fonction avec ce sommet destination en paramètre.

Remarque concernant l'algorithme d'affichage en profondeur :

Après avoir fait des modifications de dernière minute, nous avons découvert un problème avec l'implémentation de la fonction. L'affichage s'arrête juste après l'affichage du premier sommet.

Nos Tests:

Nous n'avons malheureusement pas eu le temps d'implémenter des tests rigoureux. Nous sommes pleinement conscients de l'importance des tests en programmation, et nous tenons à travailler plus sur le sujet pour les fois prochaines pour élaborer des tests qui aideront au débogage de nos programmes.

Conclusion :

Ce TP de d'algorithmique nous a permis, malgré l'erreur survenue au niveau de la fonction d'affichage en profondeur et l'absence de tests rigoureux, de consolider nos connaissances sur les graphes et leurs propriétés. Ce travail pratique nous a aussi aidé à mieux comprendre les algorithmes de parcours de graphe et l'algorithme de Dijkstra.

En conclusion, ce TP nous a aidé à mettre en œuvre les connaissances que l'on a accumulées tout au long du semestre et surtout pendant les cours d'Amphi.