

Projet : Voyageur de commerce

Alexandre Talon

2 novembre 2012

Table des matières

1	Structures de données	2
1.1	Tas	3
1.2	Arbres AVL	3
1.2.1	Principe	3
1.2.2	Rotations simples	3
1.2.3	Rotations doubles	3
1.2.4	Algorithme d'équilibrage	3
2	Algorithme de PRIM	3
2.1	Définitions	3
2.2	Algorithme	3
2.3	Implémentation	3
3	Algorithme TSP	3
3.1	Algorithme	3
3.2	Implémentation	3
4	Utilisation	3
4.1	Interface utilisateur ; compilation	3
4.2	Complexité	3

Introduction

On a ici réalisé un projet de programmation dans le langage C, qui a duré 7 semaines. Ce projet a été en très grande partie réalisée en monôme par Alexandre Talon.

On étudie un problème d'optimisation. En effet, le projet porte sur le problème du voyageur de commerce : on veut visiter un certain nombre de villes, dans le temps le plus court possible, et revenir à son point de départ. D'un point de vue plus formel, on peut voir les villes comme étant les sommets d'un graphe, la distance entre deux villes le poids de l'arête reliant les sommets correspondant, le problème étant de trouver un cycle hamiltonien de longueur minimale.

Il s'agit d'un problème NP-complet, c'est-à-dire qu'on ne dispose pas d'algorithme résolvant exactement le problème en un temps polynomial. Ainsi, l'objectif ici est de programmer un algorithme permettant de trouver une tournée passant par toutes les villes imposées par l'utilisateur, le tout en un temps raisonnable.

Pour y parvenir, il est nécessaire de s'occuper d'abord des structures de données à utiliser : quelles sont elles, à quoi serviront elles ici, quelles sont leurs avantages. Ensuite, nous verrons l'algorithme à proprement parler, découpé en fait en deux algorithmes. Enfin, nous traiterons l'aspect pratique et l'utilisation de notre programme.

1 Structures de données

On utilise dans ce projet deux structures non triviales : d'une part des tas, d'autre part des arbres AVL, qui appartiennent à la classe des arbres binaires de recherche.

1.1 Tas

1.2 Arbres AVL

1.2.1 Principe

1.2.2 Rotations simples

1.2.3 Rotations doubles

1.2.4 Algorithme d'équilibrage

2 Algorithme de PRIM

2.1 Définitions

2.2 Algorithme

2.3 Implémentation

3 Algorithme TSP

3.1 Algorithme

3.2 Implémentation

4 Utilisation

4.1 Interface utilisateur ; compilation

4.2 Complexité

Conclusion