

Algorithmique et structures de données : Mission 5

Groupe 1.2: Ivan Ahad - Jérôme Bertaux - Rodolphe Cambier
Baptiste Degryse - Wojciech Grynczel - Charles Jaquet

12 décembre 2014

Rapport écrit par Rodolphe Cambier, Ivan Ahad, Jérôme Bertaux

Introduction

Le but de la mission est de créer un algorithme capable de faire une liste de tous les coûts minimums entre chaque noeuds, le tout sur base d'une liste de paires de noeuds et du coût de leur arête.

Questions

Question 1

Les solutions proposées ne peuvent pas être uniques car il est possible qu'il existe plusieurs chemins ayant un coût minimal entre deux noeuds. Les solutions sont bien optimales car l'algorithme permet de récupérer les poids minimaux des arêtes reliant les paires de noeuds.

Question 2

Notre algorithme a une complexité supérieure à la complexité théorique en $O(m \cdot \log n)$ de l'algorithme de Kruskal où n représente le nombre d'arrêtes et m le nombre de noeuds, car la méthode `buildGraph()` de notre programme fait appel à deux boucles imbriquées, qui est appelée par la méthode "KruskaAlgorithm"

Question 3

Une technique serait de modifier l'algorithme de Kruskal, et de faire en sorte que lorsque l'on cherche la plus petite des *edges* parmi toutes celles restantes, on prenne en compte le poids des transits. Pour faire cela, il faut calculer la taille des *Clusters* correspondants à chacune des deux *vertices* aux extrémités de chaque *edge*, et les multiplier, ce qui nous donnera le nombre de transits passant par cette *edge*.

En prenant cela en compte, on peut continuer l'algorithme et trouver le graphe optimal, tenant en compte des transits.

Diagramme UML

