

Rapport d'analyse

Projet de structures de données II

Activité d'Apprentissage S-INFO-020

Membres du groupe:

LABEEUW Dorian (`dorian.labeeuw@student.umons.ac.be`)

JOERTZ Jonathan (`jonathan.joertz@student.umons.ac.be`)

Année Académique 2017-2018
Bachelier en Sciences Informatiques Bloc 3

Faculté des Sciences, Université de Mons

November 19, 2017

Abstract

Ce *rapport d'analyse* est rendu dans le cadre de l'AA S-INFO-020 "Projet de structures de données II", dispensé par le professeur *Bruyère Véronique* en année académique 2017-2018.

Contents

1	Énoncé du problème	3
2	Description des structures de données	3
2.1	Description de Q	3
2.1.1	Description	3
2.1.2	Exemple	3
2.1.3	Ordre sous-jacent	3
2.2	Description de T	3
2.2.1	Description	3
2.2.2	Exemple	3
2.2.3	Ordre sous-jacent	3
3	Explication des algorithmes présentés	3
3.1	Explication de <i>FindIntersections(S)</i>	3
3.2	Explication de <i>HandleEventPoint(p)</i>	3
3.3	Explication de <i>FindNewEvent(s_l, s_r, p)</i>	3
4	Gestion des cas particuliers	3
4.1	Gestion des segments horizontaux	3
4.2	Gestion des points d'intersection à au moins 3 segments	3
5	Description des étapes du programme	3

1 Énoncé du problème

Dans le cadre de ce projet, nous devons comprendre et implémenter une solution au problème de *map overlay* selon un article de recherche fourni. L'utilité de ce problème est de retrouver l'ensemble des points d'intersections d'un ensemble de segments donnés. Afin de réussir une implémentation efficace (on entend par là que l'on recherche une solution meilleure que la solution naïve en $O(n^2)$), il nous est nécessaire d'utiliser 2 structures de données expliquées ci-après (voir Description de Q et Description de T) et 3 algorithmes dont nous fournissons aussi une explication plus loin dans ce rapport (voir Explication de *FindIntersections(S)*, Explication de *HandleEventPoint(p)* et Explication de *FindNewEvent(s_l, s_r, p)*). De plus, certains cas poseront problème, nous donnons ici une brève explication de la façon dont nous avons prévu de les gérer (voir Gestion des segments horizontaux et Gestion des points d'intersection à au moins 3 segments). Une des applications du problème de *map overlay* se retrouve dans les cartes informatisées ou les *GPS*.

2 Description des structures de données

2.1 Description de Q

2.1.1 Description

Q est une *event queue* stockée dans un arbre binaire de recherche balancé dont l'ordre correspond à un Ordre sous-jacent défini plus loin. Les éléments stockés dans Q sont les points d'intersection entre les segments.

2.1.2 Exemple

2.1.3 Ordre sous-jacent

2.2 Description de T

2.2.1 Description

2.2.2 Exemple

2.2.3 Ordre sous-jacent

3 Explication des algorithmes présentés

3.1 Explication de *FindIntersections(S)*

3.2 Explication de *HandleEventPoint(p)*

3.3 Explication de *FindNewEvent(s_l, s_r, p)*

4 Gestion des cas particuliers

4.1 Gestion des segments horizontaux

4.2 Gestion des points d'intersection à au moins 3 segments

5 Description des étapes du programme