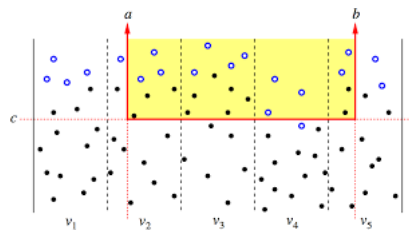




U-MONS

RAPPORT DE PROJET DE STRUCTURES DE DONNÉES 2

Priority Search Tree and Windowing



Directeurs :
G.Devillez et V.Bruyère

Groupe :
M.Salemi et A.Lecocq

16 avril 2017

Table des matières

1	Introduction	2
2	Priority Search Tree	2
2.1	Objectif de la structure	2
2.2	Définition	2
2.3	Construction	2
2.3.1	Tri des données	2
2.3.2	Création de l'arbre	2
2.3.3	Création d'un nœud	3
2.4	Windowing	3
2.5	Mise en place du problème	3
3	Idées et Mise en pratique	4
3.1	Passage des points au segments	4
4	Diagramme de Classes	4
5	Algorithmes et explications	4
5.1	Construction de l'arbre	4
6	Démonstrations	4
6.1	Arbre équilibré	4
6.2	Transformation des fenêtres	4
7	Illustrations	4
8	Mode d'emploi	5
9	Conclusion	5

1 Introduction

Dans le cadre du cours de structures de données 2, nous avons été amenés à réaliser un projet en Java. Ce projet a pour objectif de créer et manipuler une structure de données non vue au cours. Cette nouvelle structure se base sur un arbre de recherche à priorité (Priority Search Tree en anglais ou PST). De la documentation nous a été fournie afin de nous familiariser avec cette dernière structure qui elle non plus n'a pas été vue au cours.

Un PST partage certaines caractéristiques avec les tas et les Arbres Binaires de Recherche ou ABR. Ces deux dernières structures de données ayant été étudiées en profondeur lors des séances de cours, nous les considérons ici comme connues.

2 Priority Search Tree

2.1 Objectif de la structure

Un PST est une structure de données de type arbre binaire (chaque nœud comporte au plus deux fils). Cette structure de données organise des points de l'espace défini par deux coordonnées X et Y. L'organisation des données permet d'effectuer efficacement la recherche des points présents dans une fenêtre de l'espace (sans avoir à parcourir l'ensemble des points).

2.2 Définition

Un PST est une structure de données mixte. Chaque nœud est constitué d'un point et d'un nombre appelé la médiane. Si l'on considère uniquement les coordonnées X, un PST est un tas avec le minimum à la racine. Si l'on considère uniquement les coordonnées Y, le PST est un ABR avec une petite particularité. Au lieu de trier les fils par rapport à la donnée du nœud courant, un PST trie les fils en fonction de la médiane du nœud courant. Ainsi, tout nœud n d'un PST respecte les contraintes suivantes :

- son fils gauche (s'il existe ainsi que ses descendants s'ils existent) aura sa coordonnée X plus grande que celle du nœud n et sa coordonnée Y plus petite que la médiane du nœud n ;
- son fils droit (s'il existe ainsi que ses descendants s'ils existent) aura sa coordonnée X plus grande que celle du nœud n et sa coordonnée Y plus grande que la médiane du nœud n .

2.3 Construction

2.3.1 Tri des données

La construction d'un PST est plus simple si l'on construit ce dernier à partir d'une liste de points triés selon la coordonnée Y. La première étape est donc de trier les points.

2.3.2 Création de l'arbre

Pour créer un arbre, créons le nœud racine à partir de la liste des points triés.

2.3.3 Création d'un nœud

La coordonnée X d'un nœud devant être plus petite ou égale à celle de ses fils, commençons par rechercher le point avec la plus petite coordonnée X. Attribuons ce point au nœud courant. Séparons le reste des points en deux. Attribuons comme médiane du nœud courant, la moyenne de la coordonnée Y du dernier point de la partie 1 et du premier point de la partie 2. Les points étant triés selon la coordonnée Y, la première partie est disposée d'une coordonnée Y plus petite ou égale à la médiane du nœud alors que la deuxième partie aura une coordonnée Y plus grande ou égale à la médiane du nœud. Dans le cas où il ne reste aucun point après le retrait du minimum en X, la valeur de la moyenne n'a pas d'importance. Dans le cas où il ne reste qu'un point, la coordonnée Y du point restant peut être attribuée à la médiane du nœud.

2.4 Windowing

Le windowing est une technique très répandue qui consiste à sélectionner une certaine fenêtre parmi une énorme quantité de données. Un exemple pratique très répandu est l'affichage d'une carte sur un gps, le gps se voulant rapide n'affichera pas toutes les routes qui se trouvent dans le monde (énorme quantité de données) mais seulement celles qui nous entourent au moment où nous roulons avec notre véhicule.

2.5 Mise en place du problème

Pour ce projet, il nous été demandé de résoudre un problème de windowing se basant sur la structure de donnée "Priority Search Tree". Il nous est ainsi fourni un fichier.txt qui contient un ensemble de segments que nous devons pouvoir afficher et sélectionner à travers la dite technique.

Il faut donc pouvoir gérer différents type de fenêtres lors du windowing :

1. À remplir
- 2.

Il nous a aussi été fournis comme aide un extrait du livre de **à remplir**, qui nous explique la construction d'un Pst se construisant sur base de points dans le plan. La façon dont notre structure est adaptée à notre problème par rapport à cet article, ainsi que l'adaptation du windowing est expliquée dans les prochains points de ce rapport.

3 Idées et Mise en pratique

3.1 Passage des points au segments

4 Diagramme de Classes

5 Algorithmes et explications

Dans cette section, nous exposons nos différents algorithmes importants en pseudo-code ainsi que leur complexité en temps. Une courte explication en français y est aussi ajoutée afin de favoriser la compréhension de ceux-ci et leurs utilisations.

5.1 Construction de l'arbre

Algorithme 1 Construction de l'arbre

Entrée: une liste de Segment A

Sortie: /

```
1:  $max \leftarrow -\infty$ 
2: for  $i \leftarrow 1$  à  $longueur[A]$  do
3:   si  $max < A[i]$  alors
4:      $max \leftarrow A[i]$ 
5:   end si
6: end for
7: return  $max$ 
```

6 Démonstrations

Durant cette section, nous allons démontrer certaines propriétés du Pst qui ne sont pas couvertes par nos algorithmes(certaines preuves ont été réalisées lors des tests).

6.1 Arbre équilibré

6.2 Transformation des fenêtres

7 Illustrations

Ce point du rapport sert d'illustration, vous permettant d'observer différents cas d'exemples d'application du windowing, et ceci via l'application que nous avons créer.

8 Mode d'emploi

Après vous avoir montré en images notre application, nous vous expliquant ici comment faire fonctionner notre application. Vous pourrez ainsi constater cette réelle application par vous-même, qui est très intuitive et parfaitement fonctionnelle.

9 Conclusion