UMONS

STRUCTURE DE DONNÉES II
WINDOWING

Exercice Préliminaire

Élèves :
Nicolas GATTA
Arnaud SCHELLEKENS

Enseignant :
Véronique BRUYÈRE

22 décembre 2022

1 Question 1

Si on ne considère que les coordonnées en x, voyez-vous que celles-ci sont organisées selon une file à priorité ? Où est la coordonnée minimum (maximum) ? Cette file à priorité est-elle un tas ? Justifiez.

Dans un arbre de priorité qui ne considère que les coordonnées en X, les coordonnées seront organisées selon une file à priorité avec la coordonnée minimum se trouvant en tête de file et la coordonnée maximum se trouvant à la fin de la file.

Une file de priorité peut-être également considérée comme un tas. Pour rappel, un tas est une structure de données qui respecte certaines priorités : chaque élément du tas doit être plus grand (ou plus petit) que chacun de ses enfants et la racine du tas doit être l'élément avec la plus haute (ou basse) priorité.

2 Question 2

Si on ne considère que les coordonnées en y, voyez-vous que celles-ci sont organisées selon un arbre binaire de recherche ? Où est la coordonnée minimum (maximum) ? Justifiez.

Si l'on considère uniquement les données en Y sans les réorganiser, celles-ci ne sont pas organisées selon un arbre binaire de recherche, car on est pas sûr qu'en ne prenant que les Y, la règle d'avoir des valeurs plus grandes dans l'arbre gauche et plus petites dans l'arbre droit soit respectée. Cependant, en réorganisant les données, il est tout à fait possible que la règle soit respectée et alors cela sera un arbre binaire de recherche.

Si on réorganise les données, on peut avancer que la coordonnée minimum est la feuille la plus à gauche de l'arbre alors que la coordonnée maximum se trouve à la feuille la plus à droite. Sinon, si elles ne sont pas réorganisées, il faut aussi prendre en compte les racines.

3 Question 3

De quelle façon est équilibré un arbre de recherche à priorité ?

Un arbre de recherche à priorité est un arbre binaire de recherche dans lequel chaque nœud est associé à une priorité. L'équilibre dans l'arbre à priorité est géré de la même manière qu'un AVL où la balance de chaque nœud ne peut avoir que comme valeur 0, 1 , -1. On peut d'ailleurs vérifier cela grâce à la médiane du nœud racine qui donnera une différence maximum de 1 dans le sous-arbre droit et dans le sous-arbre gauche.

4 Question 4

La construction d'un arbre de recherche à priorité peut se faire en $O(n \log_2 n)$ si n est le nombre de points de $\mathbb{R}^2$ contenus dans l'arbre. Expliquez comment on peut y parvenir et comment le prouver. Il faut sans doute utiliser une autre structure de données qui permet de calculer efficacement la médiane.

Le calcul de la médiane s'effectue sur une liste de coordonnées Y triée qui sera coupée en deux. Vu que la liste est triée, les valeurs plus petites se trouveront à la gauche de la médiane dans la liste de coordonnées Y et les valeurs les plus grandes se trouveront à la droite de la médiane dans la liste de coordonnées Y. De manière générale, la complexité pour parcourir un arbre est en $O(n)$ et la hauteur de l'arbre est en $O(\log_2 n)$. Ce qui donne comme résultats $O(n \log_2 n)$.

5 Question 5

La structure d'arbre de recherche à priorité permet d'obtenir efficacement l'ensemble des points de T qui sont contenus dans une fenêtre donnée de la forme $(-\infty : x'] \times [y : y']$. Expliquez comment adapter l'algorithme proposé pour traiter une fenêtre de la forme $[x : +\infty) \times [y : y']$, $[x : x'] \times (-\infty : y']$ ou $[x : x'] \times [y : y']$.

- $[x : +\infty) \times [y : y']$: Pour ce cas, il faudrait effectuer une recherche du maximum pour les X et attribuer cette valeur à la racine pour respecter la structure de données du tas avec des valeurs cette fois plus petites en X pour chacun de ses fils.
- $[x : x'] \times (-\infty : y']$: Pour ce cas, il faudrait inverser le traitement entre X et Y. Il faut que le minimum de Y deviennent la racine et il faudrait séparer les données X en fonction de leur médiane comme pour le cas initial.
- $[x : x'] \times [y : y']$: Ce cas est une combinaison des deux précédents cas.

6 Question 6

Les auteurs du Chapitre 10 font l'hypothèse que tous les points ont des coordonnées bien distinctes en x et en y. Expliquez pourquoi.

Si les points n'étaient pas distincts, il serait impossible de respecter les conditions de construction d'un arbre de recherche à priorité, car cela pourrait poser un problème dans deux situations :
- La valeur minimum de X à la racine, car on ne sait pas comment attribuer une priorité.
- Attribuer une valeur à la médiane pour les coordonnées en Y, car il serait difficile de partager cette médiane entre plusieurs points ayant la même valeur pour la coordonnée Y.

7 Question 7

Une technique est présentée à la page 111. Celle-ci permet de simuler des coordonnées distinctes pour un ensemble de points quelconques et une requête. Expliquez comment.

Selon la technique présenter à la page 111, il faut remplacer chaque point $p := (px, py)$ par un nouveau point $\hat{p} = ((px|py), (py|px))$.
De cette manière, on obtient un nouveau point $\hat{p}$ qui a pour coordonnées X et Y des nombres composés. Grâce à tout cela, on obtiendra un nouvel ensemble de points dans lequel la première et les deuxièmes coordonnées sont uniques.

8 Question 8

Dans le chapitre, la technique présentée est adaptée à des points. Quelles différences importantes aurait-on avec des segments de droite ? Comment peut-on adapter la technique ?

Les différences entre les segments et les points sont les suivantes :

- Un segment possède deux points, chacun de ses points possédant une coordonnée X et une coordonnée Y. Ce qui veut dire qu'un segment possède deux coordonnées X et deux coordonnées Y par rapport à un point qui lui ne possède qu'une coordonnée en X et une coordonnée en Y.
- Un segment est une ligne reliant deux points, à partir du moment où deux points ne sont pas reliés, ils ne sont considérés que comme deux points séparer sans aucune corrélation.

Il faut modifier la structure pour que celle-ci accepte des segments. Le nœud contiendra alors un segment au lieu d'un point. Pour la recherche du maximum, on prendra les maximums des extrémités du segment alors que pour le minimum, on prendra les minimums des extrémités du segment. Pour la médiane, on prendra le milieu du segment.