
Devoir II

**8INF840 – Structures de données
avancées et leurs algorithmes**

Été 2017

Aymen Sioud

UNIVERSITÉ DU QUÉBEC À CHICOUTIMI
Département d'Informatique et de Mathématique

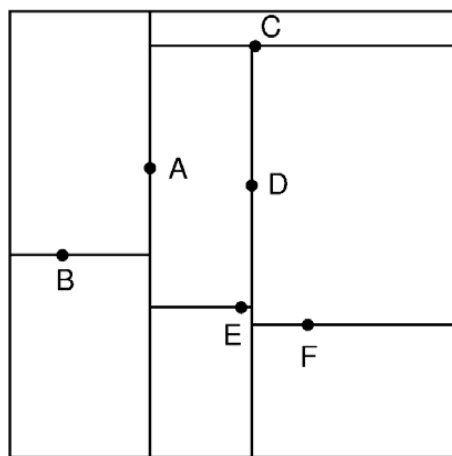
Consignes

- A remettre au plus tard le 18 juin à 18h (heure de l'est).
- Vous devrez utiliser Visual Studio 2015.
- Le travail doit être effectué en C++.
- Il ne devra pas y avoir de dossier Debug
- Le travail devra être remis via un drive par courriel à l'adresse du professeur en un seul répertoire compressé : aymen.sioud@uqac.ca.
- Le travail remis devra être nommé : Devoir2_
nom1_nom2_nom3_nom4_8INF840_E2017.

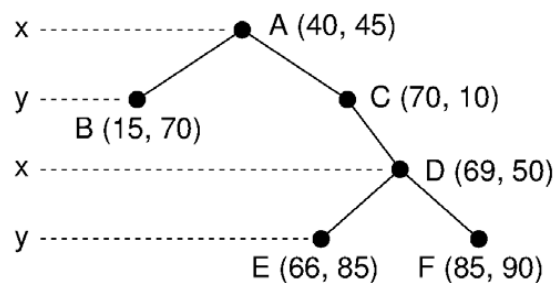
Exercice 1 : Kd-Tree

Les Kd-Tree est une structure de données qui permet de gérer le partitionnement de l'espace autour de valeurs et permet l'utilisation de nombreux algorithmes de recherche très rapide répondant à différents problèmes, tels la recherche du plus proche voisin, la recherche dans une plage de valeur.

Les arbres k-d sont des arbres binaires, dans lesquels chaque nœud contient un point en dimension k. Chaque nœud non terminal divise l'espace en deux demi-espaces. Les points situés dans chacun des deux demi-espaces sont stockés dans les branches gauche et droite du nœud courant. Par exemple, si un nœud donné divise l'espace selon un plan normal à la direction (Ox), tous les points de coordonnée x inférieure à la coordonnée du point associé au nœud seront stockés dans la branche gauche du nœud. De manière similaire, les points de coordonnée x supérieure à celle du point considéré seront stockés dans la branche droite du nœud.



(a)



(b)

On peut représenter un nœud l'arbre k-d par la classe (cette classe est une base pour plusieurs variantes) :

KDNode
«Point<K>»+Point «KDNode *»+LeftChild «KDNode *»+RigthChild
+KDNode(Point p) ~KDNode()

Il vous est demandé de coder la structure de données KD-tree avec un algorithme de recherche du plus proche voisin.

Exercice 2 : Hypergraphe et référencement web

L'indexation du web est au cœur de la navigation sur le World Wide Web. Il s'agit des différentes méthodes qui permettent de parcourir et de classer la quasi-totalité des pages web. Les moteurs de recherche utilisent des algorithmes représentant le web sous forme d'un graphe afin d'estimer la réputation de la qualité de chaque page. Chaque page est représentée par un nœud, et les liens de redirection d'une page à une autre sont représentés par un arc. Les pages ayant la meilleure réputation apparaissent alors en premier dans les résultats d'une recherche d'un utilisateur si elle correspond avec les termes de sa recherche. C'est donc un élément critique pour nombre d'acteurs du web car beaucoup d'entreprises dépendent de leur classement dans les résultats des différents moteurs de recherche.

De nombreuses stratégies d'analyse de liens ont été réalisées depuis les années 90, nous allons nous concentrer sur deux d'entre elles :

- Indegree
- PageRank qui est utilisé par le moteur de recherche de Google.

Ces deux stratégies consistent à donner un score à chaque page web. Si une page possède un score élevé, cela signifie qu'un nombre important de pages possèdent un lien vers celle-ci. On peut donc supposer qu'elle a une meilleure visibilité et donc une meilleure réputation en termes de qualité.

Une solution afin d'améliorer le classement des pages web consiste à représenter le web sous la forme d'un hypergraphe orienté plutôt qu'un graphe. Cela permet de limiter la redondance des connexions, et donc d'améliorer le classement obtenu.

A partir du graphe de base, les nœuds (des pages web), sont regroupés en blocs non séquentiels. Les arêtes (les liens) sont utilisées pour construire les hyperarcs. Un hyperarc relie un bloc B à une page p si et seulement si au moins une des pages du bloc pointe vers p.

On ne prend pas en compte les liens d'une page vers elle-même dans le processus.

L'intérêt de regrouper les pages par blocs est de ne représenter que les connexions indépendantes. En effet, avec un graphe simple, si toutes les pages d'un bloc pointent vers une même autre page, on a alors autant d'arêtes que de pages dans le bloc, alors que ces pages peuvent par exemple avoir été créés par la même personne.

Le résultat de la représentation obtenue avec un hypergraphe dépend de la granularité choisie pour les blocs : plus celle-ci est fine (un bloc = une page), plus une page a de l'influence. En jouant sur la granularité, on peut ainsi limiter l'influence des pages dépendantes, rendre les liens plus pertinents. Il est nécessaire de choisir cette granularité selon l'objectif. En effet, plus celle-ci est élevée, plus le nombre d'hyperarcs est faible. Une granularité trop grande entraînera donc un nombre trop petit, et inversement, ce qui peut fausser la réputation des pages. On considère ici trois niveaux de granularité, basés sur l'URL des pages (faible coût de calcul) :

- Regroupement par page : un bloc est composé d'une seule page web. Cela revient à utiliser le modèle de base.
- Regroupement par domaine : un bloc est composé de toutes les pages appartenant à un même domaine.
- Regroupement par hôte : un bloc est composé de toutes les pages d'un même hôte.

Il vous est demandé de mettre en place et d'implémenter les deux algorithmes cités plus hauts en utilisant les hypergraphes. 2 datasets pour vos tests sont disponibles.

Exercice 3 : Parcours de Graphes

En 2020, un virus transforma une grosse partie de la population en zombies. La plupart des villes furent isolées et une grosse partie de la population fut décimée. Afin d'éliminer le maximum de zombies, la résistance mis en place un plan. Plusieurs tours de tirs ont été placées à des endroits stratégiques afin de maximiser le nombre de zombies tués.

Actuellement vous avez un problème d'approvisionnement en cartouches sur plusieurs des tours. Vous devez approvisionner toutes les tours de tir avec la distance minimale. Tous les points d'accès sont possibles.

Concevez et coder un algorithme qui permettra d'approvisionner toutes les tours en donnant le meilleur point de début.

Utiliser le fichier ville.txt joint qui contient les coordonnées euclidiennes.