Rapport : SDA City Search

Membre du groupe

Nom Prénom	Numéro Etudiant	Pseudonyme
Bayard Emeric	11606611	Dryska
Sheikh Rakib	11502605	NoobZik
Yesli Rayane	11507199	soso7
Belmaati Yacine	11513398	Laser1W

Analyse théorique et empirique

1) Choix d'implémentation

2) Pseudo-code de getInRange

```
# Prends en paramètre : BinarySearchTree *, void *keyMin, void *keyMax
# 2 LinkedList (LL and File)
# LL for returning a filtered city
# file for Level Order Transversal BinarySearchTree

WHILE temp exist
  if minimum < temp->key AND temp->key < maximum
    if insertInLinkedList(LL, temp->value) == false;
      ret NULL;
    if temp->left exist
      if insertInLinkedList(File, temp->left) == false;
        ret NULL
    if temp->right exist
      if insertInLinkedList(File, temp->right) == false;
        ret NULL;
  temp = extractFile(File);
freeLinkedList (File);
```

3) Analyse de compléxité

- insertInLinkedList à une compléxité de *θ(1)*, on l'ajoute directement à la fin de la liste
- extractFile à une compléxité de *θ(1)* aussi, elle retire le premier élément de la liste.
- Les instructions if sont de compléxité *θ(1)*.
- **En répétant N fois cette boucle, on a donc une compléxité de *θ(n)***.
- Dans le meilleur cas : *θ(1)*.
- Dans le pire cas : *θ(n)*

4) Pseudo-code de intersect

```
tmpA = listA->head;
tmpB = listB->head;
while tmpA exist
  while tmpB exist
    if tmpA->value == tmpB->value
      if insertInLinkedList(listC, tmpA->value) == false;
        ret NULL;
    tmpB = tmpB->next
  tmpA = tmpA->next;
  tmpB = listB->head;
ret listC;
```

5) Analyse de compléxité

Dans le pire cas, soit M = N, on aura une compléxité de M à la puissance N. On peut dire que la manière dont l'algorithme est écrite

est très lente. Dans le meilleur cas, on aura $\theta(1)$

6) Comparaison des 3 approches

Le code Morton est érroné, elle renvoie plus de ville que l'algorithme de base, c'est à dire, l'algorithme sur les Listes uniquement.

Pour calculer le temps on décide de le faire sur 1 000 ville et 1 000 000 ville avec les commandes suivantes :

```
$ time ./boxsearch cities_1000.csv 1 1 10 10
$ time ./boxsearch cities_10000000.csv 1 1 10 10
```

Pour latitude 1 - 10 et longitude 1 - 10 pour 1 000 villes

	Temps écoulé
Algorithme 1	Item Two
Algorithme 2	
Algorithme 3	Erreur de ville