Rapport : Projet de Programmation II

Introduction :

Durant ce projet on se devait de résoudre le problème du traversier en parties différentes la première en utilisant une BigTable, et la deuxième en utilisant une HashTable.

Partie 1 :

La capture d'écran ci-dessous et les fichiers textes output dans le sous-dossier BigTable confirme que notre solution en utilisant une BigTable fonctionne parfaitement, la soumission en ligne peut nous le confirmer dans ce rapport.

My Submissions

#	Problem	Verdict	Language	Run Time	Submission Date
25824044	10261 Ferry Loading	Accepted	JAVA	0.650	2020-12-08 01:42:44
25824042	10261 Ferry Loading	Accepted	JAVA	0.710	2020-12-08 01:42:37
25824040	10261 Ferry Loading	Accepted	JAVA	0.710	2020-12-08 01:42:30
25824039	10261 Ferry Loading	Accepted	JAVA	0.710	2020-12-08 01:42:22

Soumission en ligne de la solution avec BigTable

Partie 2 :

La capture d'écran ci-dessous et les fichiers textes output dans le sous-dossier HashTable confirme que notre solution en utilisant une HashTable fonctionne parfaitement, la soumission en ligne peut nous le confirmer dans ce rapport.

25824024	10261 Ferry Loading	Accepted	JAVA	0.590	2020-12-08 01:33:18
25824023	10261 Ferry Loading	Accepted	JAVA	0.520	2020-12-08 01:32:59
25824021	10261 Ferry Loading	Accepted	JAVA	0.280	2020-12-08 01:32:44
25824019	10261 Ferry Loading	Accepted	JAVA	0.500	2020-12-08 01:31:52

Figure 2 : Soumission en ligne de la solution avec HashTable

Design de la partie 2:

Au début, j'entrais des clés aléatoirement mais ça avait toujours le même résultat : : 'run time error '. Pour résoudre le problème j'ai enlevé l'initialisation de toutes les valeurs de la Hashmap en false, et j'ai travaillé avec null comme étant une valeur de défaut car si la valeur est toujours null donc le cas n'a toujours pas été visite. Âpres que n'importe quelle fonction marchait, et suite à plusieurs tentatives je suis arrive à la fonction que j'ai soumis qui consiste à incrémenter currS de 1 ou 10 ou 100 selon la valeur de k modulo 3.
Cette fonction de hachage mas permis d'avoir un temps d'exécution dans les alentours de 0.5s.

Conclusion :

En conclusion, on peut dire que les deux solutions sont viables, et que le temps d'exécution de la Hashtable dépend de la fonction de hachage choisie.