

ALGORITHMIQUE AVANCEE TP N°1

Objectif du TP : Se re-familiariser avec le langage java, comprendre l'intérêt des tables de hachage. Les TP doivent être finis avant le début de la séance suivante.

ENONCE :

L'objectif de ce TP est de construire un programme qui pour chaque mot français (conjugué) soit capable d'indiquer quel est le mot (non conjugué) associé. L'ensemble des associations existantes est contenu dans le fichier "dico.txt" qui contient sur chaque ligne un mot et le mot dont il est issu. Ce fichier compte a peu près 20000 mots.

Nous voulons comparer différentes implémentations permettant de manipuler cette table associative.

- *Dans la première implémentation (fournie sur le site du cours) , les mots et leur correspondance sont stockés dans une liste chaînée.*
- *Dans la seconde implémentation (fournie sur le site du cours), les mots et leurs correspondances sont stockés dans une Table de Hachage, qui utilise les librairies Java.*
- *Dans la troisième implémentation (que vous devrez faire), les mots et leurs correspondances seront stockés dans une Table de Hachage que vous créerez de toutes pièce.*

Pour chaque version, il y aura un fichier DicoXXX.java qui implémente la classe dictionnaire (XXX décrira l'implémentation choisie) et un fichier TestDicoXXX.java qui permettra de mesurer les temps de calculs pour une recherche dans le dictionnaire.

Exercice 1 : La version Liste chaine (2 points)

Regardez le code des fichiers DicoLinkedList.java et TestDicoLinkedList.java .Assurez vous de comprendre chaque ligne du code, posez des questions si quelque chose vous échappe.

Compilez et exécutez le programme TestDicoLinkedList.

Que constatez vous ?

Exercice 2 : La version Hashtable de l'API java (3 points)

Regardez le code des fichiers DicoHashtable.java et TestDicoHashtable.java . Assurez vous de comprendre chaque ligne du code, posez des questions si quelque chose vous échappe.

Compilez et exécutez le programme TestDicoHashtable

Que constatez vous ? Comparez avec la version précédente, expliquez pourquoi les résultats diffèrent. Que se passerait il si on augmentait la taille du dictionnaire pour la faire passer à 40000 mots ?

Exercice 3 : La version Hashtable de votre création (15 points)

Faites une classe HashtableLicence permettant de mettre en oeuvre les méthodes dont vous avez besoin pour faire les tests. Faites également le programme de test associé. Comparez avec les versions précédentes.

Remarque : Vous pouvez utiliser la fonction hashCode de la classe String.