

Sujet : Recherche dichotomique dans un fichier structuré trié.

Première étape :

Il s'agit, à partir de données stockées dans un fichier texte de générer un autre fichier où ces mêmes données seront écrites sous une forme structurée de manière à pouvoir être manipulées globalement ensuite, et surtout on pourra faire de l'accès direct sur ce second fichier.

Pour ces données, chaque "individu" ou élément est constitué de plusieurs informations, certaines étant alphanumériques et d'autres numériques.

Pour simplifier le problème on considérera que le fichier de départ est correct, donc ne contient pas d'erreur, et on n'aura pas à vérifier l'exactitude des données.

Deuxième étape :

A partir du fichier structuré généré à l'étape précédente, il s'agit de trier ses éléments par ordre alphabétique sur les noms et de les réécrire dans le même fichier, mais triés cette fois-ci. Le nombre d'éléments du fichier n'étant pas très important, on pourra se contenter d'une méthode de tri peu performante : le tri par extraction dans sa version élémentaire par exemple.

Troisième étape :

A partir du fichier structuré et trié par ordre alphabétique résultant des deux étapes précédentes, il s'agit maintenant de pouvoir retrouver un élément de ce fichier, à partir de son nom, par une méthode dichotomique.

Le principe d'une telle méthode est le suivant :

- On compare l'élément recherché à l'élément médian du fichier ; s'il est plus petit, alors cela signifie que l'élément cherché se trouve dans la première moitié du fichier ; s'il est plus grand, alors il se situe dans la seconde moitié ; bien entendu si on n'est dans aucun de ces deux cas, l'élément cherché est l'élément considéré et on a terminé la recherche.

- Si lors de la comparaison précédente, on est dans l'un des deux premiers cas, alors on recommence la même opération dans la moitié du fichier qui doit être prise en compte ; et cela jusqu'à ce qu'on ait trouvé l'élément souhaité pour lequel on affichera à l'écran toutes les informations figurant dans le fichier.

Une telle méthode de recherche d'un élément dans un ensemble trié d'éléments est performante, beaucoup plus qu'une méthode séquentielle.

Application : un salon des vins.

Il s'agit de pouvoir retrouver rapidement un exposant de ce salon à partir du nom du vin qu'il présente à ce salon, par exemple "Château Margaux", afin d'en obtenir certaines caractéristiques complémentaires.

Les données du fichier de départ sont organisées de la façon suivante :

- il y a un vin par ligne

- pour chaque vin on a les informations suivantes, dans l'ordre : nom du vin, appellation, région, nom du propriétaire, surface en hectares, numéro de stand ; deux informations consécutives sont séparées par une tabulation.

En tout premier lieu vous définirez la structure de données à utiliser, puis vous écrirez un algorithme correspondant à chacune des trois étapes définies précédemment, dans le cadre de l'application proposée, en pseudo-code. Ensuite vous écrirez un programme modulaire en langage Java, correspondant à cette application. Bien entendu, pour les deux premières étapes, il faudra vérifier le contenu du fichier obtenu en affichant les éléments après traitement.