Question 2

Détermination de X à partir d’une méthode d’erreur :

Afin de déterminer le X (la valeur suffisamment grande pour laquelle il faut plus de 10s au tableau pour se remplir), la méthode d’erreur prend en paramètre la taille du tableau à tester et mesure le temps qu’elle prend pour faire l’addition des valeurs dans le tableau . Si le temps est inferieur est inférieur à 10 secondes, la fonction retourne false sinon elle retourne vraie.

200 000 000 est la valeur que nous avons retenue pour le ArrayList.

De la même façon nous avons déterminé des valeurs pour les autres structures de données.

**public** **static** **Boolean** arraylistFillRuntime(**int** sizeArray ) // recuperer avec sizeArray la taille a tester

{

List <Character> tableau = **new** ArrayList<Character>();// creation d’un tableau dynamique Arraylist

**double** heure = (**double**) (System.*currentTimeMillis*()/1000); // Mesure du temps initial

// remplir le tableau

**for** (**int** i=0; i< sizeArray; i++)

{

tableau.add(*random\_char*());

}

//prendre l'heure a la fin //determiner le temps que ca a pris

**double** heure\_fin = (**double**) (System.*currentTimeMillis*()/1000);

heure = heure\_fin - heure ;

// comparer le temps ecoule avec 10s ;

**if** (heure < 10)

**return** **false** ;

**else**

**return** **true**;

}

**Résultats des comparaisons**

Afin de réaliser les comparaisons, nous avons remplis tous les tableaux (Arrays, Arraylist, Linkedlist) d’avec les mêmes nombres de valeurs et comparé le temps nécessaire pour l’addition des données dans chacune des structures de données.

Etant donné que la structure de données LinkedList requiert beaucoup plus de place, la comparaison avec elle s’arrête a une certaine valeur « plafond » après laquelle, un essai d’ajout entraine une erreur.

Le résultat se présente comme suit :

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Nombre d'elements | ArrayList | LinkedList | Tableau | Array/Linked | Array/Tableau | Linked/Tableau |
| 25 000 000 | 3294 | 4784 | 1184 | 0.247 |  |  |
| 50 000 000 | 3626 | 10840 | 2366 |  |  |  |
| 75 000 000 | 5312 | 16570 | 3700 |  |  |  |
| 100 000 000 | 8023 | 22600 | 4963 |  |  |  |
| 125 000 000 | 9649 | 34121 | 5982 |  |  |  |
| 200 000 000 | 10629 | S/O | 7250 |  |  |  |
| 300 000 000 | 21410 | S/O | 14212 |  |  |  |
| 350 000 000 | 24971 | S/O | 16620 |  |  |  |

Nous constatons que l’addition des éléments se fait plus vite lorsqu’on utilise des tableaux simples. Par conséquent, un concepteur qui connait déjà la plage de données avec laquelle il doit travailler aurait tout intérêt à utiliser un tableau pour stocker ses données.

Cependant si, pour un besoin de flexibilité, un tableau dynamique est le meilleur choix , alors la structure de données à privilégier est la ArrayList.