

Pour cet exercice, nous devons comparer les algorithmes de recherche séquentielle et de recherche par dichotomie sur leur performances. Pour ce faire, nous disposons d'une classe Recherche, contenant nos deux algorithmes, une méthode trier qui permet de trier un tableau de chaîne de caractères et une méthode genererMots permettant de créer un tableau de chaînes de caractères de taille 2. Nous faisons nos testes dans une classe Test qui génère un tableau de caractères, grâce a genererMots, le tri et recherche une même chaîne de caractères créée grâce a genererMots avec nos deux algorithmes. Il mesure le temps avant et après chaque recherche afin d'en déduire le temps d'exécution et fini par nous afficher la chaîne de caractères recherchée et le temps d'exécution de nos deux algorithmes.

Les tests doivent être effectués sur la même machine afin d'obtenir des résultats homogènes (performances équivalentes).

De plus, il ne faut pas avoir de programmes autres que le terminal et activer le mode avion de son ordinateur pour ne pas recevoir de données extérieures pouvant fausser les résultats, afin que l'ordinateur ne soit occupé qu'à trier les valeurs (et donc qu'il soit à son niveau de performance optimal).

	1 000 valeurs	10 000 valeurs	100 000 valeurs
recherche séquentielle	11745	66372	3543290
Recherche dichotomie	1920	2476	4206

On peut observer que la recherche Séquentielle est très lente. On explique cette lenteur par le fait qu'elle parcourt toute la liste, à l'inverse, la recherche dichotomique cherche à découper la liste en deux jusqu'à trouver la valeur recherchée, ce qui fait qu'elle ne fait que quelques comparaisons sur un tableau de plusieurs milliers de valeurs.

```
gt220862@c-di-712-10:~/TP/dossier_exam$ javac Test.java
gt220862@c-di-712-10:~/TP/dossier_exam$ java Test
Les recherches doivent trouver : HK sur une plage de 1000 valeurs
La recherche séquentielle a trouvée HK en 9237
La recherche dichotomique a trouvée HK en 1608
```