

Laboratoire 04 (partie 1) – Fouilles et tris

OBJECTIFS :

- Continuer à vous exercer à coder en Java, en respectant les bonnes pratiques et les normes de programmation (style Java).
- Approfondir votre compréhension de la fouille binaire en codant son algorithme, en Java, tout en vous exerçant à manipuler les tableaux.
- Approfondir votre compréhension des tris (sélection, insertion, bulles) en codant leur algorithme, en Java, tout en vous exerçant à manipuler les tableaux.

Avant de commencer, importez dans votre projet les classes fournies `UtilitaireFouilles.java` et `UtilitaireTris.java`.

EXERCICE 1

1. La classe `UtilitaireFouilles` contient déjà les deux méthodes qui implémentent la fouille séquentielle non ordonnée, et la fouille séquentielle ordonnée. Vous devez implémenter la méthode `fouilleBinaire` dont la coquille se trouve déjà dans la classe `UtilitaireFouilles`. Référez-vous à l'algorithme qu'on a vu en classe.
2. Ajoutez ensuite une instruction qui affiche l'élément milieu, à chaque passe, pour qu'on puisse voir les éléments consultés lors d'une fouille binaire.
3. Créer une autre classe, avec une méthode `main`, pour tester votre méthode.
 - a. Créer un tableau qui contient les éléments suivants :
[3, 5, 6, 8, 12, 13, 16, 20, 22, 24, 27, 31, 33, 35, 43, 46, 51, 53, 56, 58, 60, 63, 65, 67, 81]
 - b. Faites une **recherche** binaire du nombre **53** dans le tableau.
 - i. Les éléments consultés par la recherche devraient être : **33 56 46 51 53** .
 - ii. La méthode devrait retourner l'indice **17** .
 - c. Faites une **recherche** binaire du nombre **17** dans le tableau.
 - i. Les éléments consultés par la recherche devraient être : **33 13 22 16 20** .
 - ii. La méthode devrait retourner l'indice **-1** (pas dans le tableau).

EXERCICE 2

1. La classe `UtilitaireTris` contient la méthode `permuter`, qui permet de permuter deux éléments d'un tableau d'entiers, et la méthode `afficher`, qui permet d'afficher les éléments d'un tableau. Votre travail consiste à implémenter tous les algorithmes de tri qu'on a vus en classe (insertion, sélection, bulles). N'oubliez pas d'utiliser la méthode `permuter` lorsque possible. La classe `UtilitaireTris` contient déjà les coquilles de ces méthodes, que vous devez compléter.

Chaque fois que vous avez codé une méthode, testez-la, et déboguez-la tout de suite, avant de passer au codage d'une autre.

2. Pour tester vos méthodes, créer une autre classe, avec une méthode `main`. Créez divers tableaux d'entiers que vous enverrez comme paramètres effectifs à l'appel de vos méthodes de tri. Pour chaque appel, affichez le tableau avant le tri, et le tableau après le tri (avec la méthode `afficher`), et vérifiez qu'il est bien trié.

BON TRAVAIL !