

# Algorithmes de recherche en Java

Rechercher consiste à trouver des données de manière efficace. Explorons la recherche binaire et la recherche linéaire.

## 1. Recherche binaire : Efficacité logarithmique

La recherche binaire fonctionne sur des tableaux triés, réduisant l'espace de recherche de moitié à chaque étape, avec un temps de  $O(\log n)$ .

### Implémentation en Java

```
public class BinarySearch {  
    public static int binarySearch(int[] arr, int target) {  
        int left = 0, right = arr.length - 1;  
        while (left <= right) {  
            int mid = left + (right - left) / 2;  
            if (arr[mid] == target) return mid;  
            if (arr[mid] < target) left = mid + 1;  
            else right = mid - 1;  
        }  
        return -1;  
    }  
  
    public static void main(String[] args) {  
        int[] arr = {2, 3, 4, 10, 40, 50};  
        int target = 10;  
        System.out.println("Trouvé à : " + binarySearch(arr, target));  
    }  
}
```

**Sortie :** Trouvé à : 3

## 2. Recherche linéaire : Base simple

La recherche linéaire vérifie chaque élément séquentiellement, avec un temps de  $O(n)$ —utile pour les données non triées.

## Implémentation en Java

```
public class LinearSearch {  
    public static int linearSearch(int[] arr, int target) {  
        for (int i = 0; i < arr.length; i++) {  
            if (arr[i] == target) return i;  
        }  
        return -1;  
    }  
  
    public static void main(String[] args) {  
        int[] arr = {10, 20, 80, 30, 60, 50, 110};  
        int target = 30;  
        System.out.println("Trouvé à : " + linearSearch(arr, target));  
    }  
}
```

**Sortie :** Trouvé à : 3