

Searching Algorithms in Java

Searching is about finding data efficiently. Let's explore binary search and linear search.

1. Binary Search: Logarithmic Efficiency

Binary search works on sorted arrays, reducing the search space by half each step, with $O(\log n)$ time.

Java Implementation

```
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("Found at: " + binarySearch(arr, target));  
    }  
}
```

Output: Found at: 3

2. Linear Search: Simple Baseline

Linear search checks every element sequentially, with $O(n)$ time—useful for unsorted data.

Java Implementation

```
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("Found at: " + linearSearch(arr, target));
}
}
```

Output: Found at: 3