



[Data Structures](#)

Algorithms

[algorithm visualizer](#)

<https://github.com/TheMIU/Algorithms>

Commonly used Algorithms

- Sorting Algorithms
 - Bubble Sort
 - Selection Sort
- Search Algorithms
 - Linear Search
 - Binary Search
- Graph Algorithms
- Tree Algorithms
- Dynamic Programming Algorithms

Sorting Algorithms

[Sorting Algorithms Animations | Toptal®](#)

- ▶ 15 Sorting Algorithms in 6 Minutes
- ▶ Sorting Algorithms Explained Visually

Bubble sort

- ▶ Bubble sort in 2 minutes

```
for i from 1 to N
  for j from 0 to N - 1
    if a[j] > a[j + 1]
      swap ( a[j] , a[j + 1] )
```

```

public class BubbleSort {
    public static void sort(int[] arr) {
        int n = arr.length;

        for (int i = 0; i < n - 1; i++) {
            for (int j = 0; j < n - i - 1; j++) {
                if (arr[j] > arr[j + 1]) {
                    // swap
                    int temp = arr[j];
                    arr[j] = arr[j + 1];
                    arr[j + 1] = temp;
                }
            }
        }
    }
}

```

Selection Sort

📺 Selection sort in 3 minutes

```

for (j = 0; j < n-1; j++)
    int iMin = j;

    for (i = j+1; i < n; i++)
        if (a[i] < a[iMin])
            iMin = i;

    if (iMin != j)
        swap(a[j], a[iMin]);

```

```

public class SelectionSort {
    public static void sort(int[] arr) {
        int n = arr.length;

        for (int i = 0; i < n - 1; i++) {
            int iMin = i;

            // Find the index of the minimum element
            for (int j = i + 1; j < n; j++) {
                // Compare elements
                if (arr[j] < arr[iMin]) {
                    iMin = j;
                }
            }

            if (iMin != i) {
                // Swap arr[i] and arr[iMin]
                int temp = arr[i];
                arr[i] = arr[iMin];
                arr[iMin] = temp;
            }
        }
    }
}

```

Search Algorithms

Linear Search

```

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 the index if found
    }
}
return -1; // Return -1 if not found
}
}

```

Binary Search

📺 Binary search in 4 minutes

```

public class BinarySearch {
    public static int binarySearch(int[] arr, int target) {
        int left = 0;
        int 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; // Not found
    }
}

```