

BASIC SORTING ALGORITHM

1. Bubble Sort (*increasing order*)

```
import java.util.Scanner;

public class bubbleSort { // O(n^2)

    public static void bubble_sort (int arr[]) {
        for (int turn = 0; turn < arr.length-1; turn++) {
            for (int j = 0; j < arr.length-1-turn; j++) {
                if (arr[j] > arr[j+1]) {
                    // swapping
                    int temp = arr[j];
                    arr[j] = arr[j+1];
                    arr[j+1] = temp;
                }
            }
        }
    }

    public static void printArr (int arr[]) {
        for (int i = 0; i < arr.length; i++) {
            System.out.print(arr[i] + " ");
        }
        System.out.println();
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        System.out.print("Enter size of an Array: ");
        int size = sc.nextInt();
        int num[] = new int[size];
        System.out.print("Enter elements of an Array: ");
        for (int i = 0; i < num.length; i++) {
            num[i] = sc.nextInt();
        }
        bubble_sort(num);
        System.out.print("Sorted Array: ");
        printArr(num);
    }
}
```

2. Bubble Sort (*decreasing order*)

```
public class BubblesSort {

    public static void printArr(int arr[]) {
        for (int i = 0; i < arr.length; i++) {
            System.out.print(arr[i] + " ");
        }
        System.out.println();
    }

    public static void Bubble_sort(int arr[]) {
        for (int turn = 0; turn < arr.length-1; turn++) {
            for (int j = 0; j < arr.length-1-turn; j++) {
                if (arr[j] < arr[j+1]) {
                    int temp = arr[j];
                    arr[j] = arr[j+1];
                    arr[j+1] = temp;
                }
            }
        }
    }

    public static void main(String[] args) {
        int arr[] = {3, 6, 2, 1, 8, 7, 4, 5, 3, 1};
        Bubble_sort(arr);
        printArr(arr);
    }
}
```

3. Counting Sort (*increasing order*)

```
import java.util.Scanner;
public class countingSort {
    public static void printArr (int arr[]) {
        for (int i = 0; i < arr.length; i++) {
            System.out.print(arr[i] + " ");
        }
        System.out.println();
    }
    public static void counting_sort (int arr[]) {
        int largest = Integer.MIN_VALUE;
        for (int i = 0; i < arr.length; i++) {
            largest = Math.max(largest, arr[i]);
        }
        int count[] = new int[largest+1]; // +1 becoz we're starting from 0
        for (int i = 0; i < arr.length; i++) {
            count[arr[i]]++;
        }
        int j = 0;
        for (int i = 0; i < count.length; i++) {
            while (count[i] > 0) {
                arr[j] = i;
                j++;
                count[i]--;
            }
        }
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        System.out.print("Enter size of the Array: ");
        int size = sc.nextInt();
        int arr[] = new int[size];
        System.out.print("Enter elements of an Array: ");
        for (int i = 0; i < arr.length; i++) {
            arr[i] = sc.nextInt();
        }
        counting_sort(arr);
        System.out.print("Sorted Array: ");
        printArr(arr);
    }
}
```

4. Counting Sort (*decreasing order*)

```
public class SelectionsSort {
    public static void printArr (int arr[]) {
        for (int i = 0; i < arr.length; i++) {
            System.out.print(arr[i] + " ");
        }
        System.out.println();
    }
    public static void Selection_sort (int arr[]) {
        for (int i = 0; i < arr.length; i++) {
            int minPos = i;
            for (int j = i+1; j < arr.length; j++) {
                if (arr[minPos] < arr[j]) {
                    minPos = j;
                }
            }
            int temp = arr[minPos];
            arr[minPos] = arr[i];
            arr[i] = temp;
        }
    }
    public static void main(String[] args) {
        int arr[] = {3, 6, 2, 1, 8, 7, 4, 5, 3, 1};
        Selection_sort(arr);
        printArr(arr);
    }
}
```

5. Insertion Sort (*increasing order*)

```
import java.util.Scanner;

public class insertionSort {

    public static void printArr (int arr[]) {
        for (int i = 0; i < arr.length; i++) {
            System.out.print(arr[i] + " ");
        }
        System.out.println();
    }

    public static void insertion_sort (int arr[]) {
        for (int i = 1; i < arr.length; i++) {
            int curr = arr[i];
            int prev = i-1;
            // Finding out the right position to do insertion
            while (prev >= 0 && arr[prev] > curr) {
                arr[prev+1] = arr[prev];
                prev--;
            }
            // Insertion
            arr[prev+1] = curr;
        }
    }

    public static void main (String args[]) {
        Scanner sc = new Scanner (System.in);
        System.out.print("Enter size of an Array: ");
        int size = sc.nextInt();
        int arr[] = new int[size];
        System.out.print("Enter elements of an Array: ");
        for (int i = 0; i < arr.length; i++) {
            arr[i] = sc.nextInt();
        }
        insertion_sort(arr);
        System.out.print("Sorted Array: ");
        printArr(arr);
    }
}
```

6. Insertion Sort (*decreasing order*)

```
public class InsertionsSort {

    public static void printArr(int arr[]) {
        for (int i = 0; i < arr.length; i++) {
            System.out.print(arr[i] + " ");
        }
        System.out.println();
    }

    public static void Insertion_sort(int arr[]) {
        for (int i = 1; i < arr.length; i++) {
            int curr = arr[i];
            int prev = i - 1;
            while (prev >= 0 && arr[prev] < curr) {
                arr[prev+1] = arr[prev];
                prev--;
            }
            arr[prev+1] = curr;
        }
    }

    public static void main (String[] args) {
        int arr[] = {3, 6, 2, 1, 8, 7, 4, 5, 3, 1};
        Insertion_sort(arr);
        printArr(arr);
    }
}
```

7. Selection Sort (*increasing order*)

```
import java.util.Scanner;

public class selectionSort {

    public static void printArr (int arr[]) {
        for (int i = 0; i < arr.length; i++) {
            System.out.print(arr[i] + " ");
        }
        System.out.println();
    }

    public static void selection_sort (int arr[]) {
        for (int i = 0; i < arr.length-1; i++) {
            int minPos = i;
            for (int j = i+1; j < arr.length; j++) {
                if (arr[minPos] > arr[j]) { // if we want it in decreasing order then change > to <
                    minPos = j;
                }
            }
            int temp = arr[minPos];
            arr[minPos] = arr[i];
            arr[i] = temp;
        }
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        System.out.print("Enter size of an Array: ");
        int size = sc.nextInt();
        int arr[] = new int[size];
        System.out.print("Enter elements of an Array: ");
        for (int i = 0; i < arr.length; i++) {
            arr[i] = sc.nextInt();
        }
        selection_sort(arr);
        System.out.print("Sorted Array: ");
        printArr(arr);
    }
}
```

8. Selection Sort (*decreasing order*)

```
public class SelectionsSort {

    public static void printArr (int arr[]) {
        for (int i = 0; i < arr.length; i++) {
            System.out.print(arr[i] + " ");
        }
        System.out.println();
    }

    public static void Selection_sort (int arr[]) {
        for (int i = 0; i < arr.length; i++) {
            int minPos = i;
            for (int j = i+1; j < arr.length; j++) {
                if (arr[minPos] < arr[j]) {
                    minPos = j;
                }
            }
            int temp = arr[minPos];
            arr[minPos] = arr[i];
            arr[i] = temp;
        }
    }

    public static void main(String[] args) {
        int arr[] = {3, 6, 2, 1, 8, 7, 4, 5, 3, 1};
        Selection_sort(arr);
        printArr(arr);
    }
}
```