

# 1. Sorting and Searching Techniques

## 1. Bubble Sort

### Code:

```
import java.util.Scanner;

public class BubbleSort {

    public static void bubbleSort(long[] list) {

        int size = list.length;

        for (int hold = 0; hold < size - 1; hold++) {

            boolean flag = false;

            for (int walker = 0; walker < size - hold - 1; walker++) {

                if (list[walker] > list[walker + 1]) {

                    long t = list[walker];

                    list[walker] = list[walker + 1];

                    list[walker + 1] = t;

                    flag = true;

                }

            }

            if (!flag) break;

            System.out.print("Pass:" + (hold + 1) + " : ");

            for (int i = 0; i < size; i++) System.out.print(list[i] + " ");

            System.out.println();

        }

    }

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the size of array: ");
```

```

        int size = scanner.nextInt();

        long[] num = new long[size];

        System.out.print("Enter the elements: ");

        for (int i = 0; i < size; i++) num[i] = scanner.nextLong();

        System.out.print("Unsorted Array: ");

        for (int i = 0; i < size; i++) System.out.print(num[i] + " ");

        System.out.println();

        bubbleSort(num);

        System.out.print("Sorted Array: ");

        for (int i = 0; i < size; i++) System.out.print(num[i] + " ");

        scanner.close();

    }

}

```

### Output :

```

Enter the size of array: 5
Enter the elements: 64 34 25 12 22

Unsorted Array: 64 34 25 12 22

Pass:1 : 34 25 12 22 64
Pass:2 : 25 12 22 34 64
Pass:3 : 12 22 25 34 64

Sorted Array: 12 22 25 34 64

```

## 2. Insertion Sort

### Code:

```

import java.util.Scanner;

public class InsertionSort {

    public static void insertionSort(int[] list) {

```

```

int n = list.length;

for (int key = 1; key < n; key++) {

    int walker = key - 1;

    int hold = list[key];

    while (walker >= 0 && hold < list[walker]) {

        list[walker + 1] = list[walker];

        walker--;

    }

    list[walker + 1] = hold;

    System.out.print("\nPass : " + key + " ");

    for (int i = 0; i < n; i++) System.out.print(list[i] + " ");

}

}

public static void main(String[] args) {

    Scanner scanner = new Scanner(System.in);

    System.out.print("Enter the size of array: ");

    int n = scanner.nextInt();

    int[] num = new int[n];

    System.out.print("Enter the elements: ");

    for (int i = 0; i < n; i++) num[i] = scanner.nextInt();

    System.out.print("\nUnsorted Array: ");

    for (int i = 0; i < n; i++) System.out.print(num[i] + " ");

    System.out.println();

    insertionSort(num);

    System.out.print("\nSorted Array: ");

    for (int i = 0; i < n; i++) System.out.print(num[i] + " ");

```

```

        scanner.close();
    }
}

```

```

Enter the size of array: 5
Enter the elements: 64 34 25 12 22

Unsorted Array: 64 34 25 12 22

Pass : 1   34 64 25 12 22
Pass : 2   25 34 64 12 22
Pass : 3   12 25 34 64 22
Pass : 4   12 22 25 34 64
Sorted Array: 12 22 25 34 64

```

#### 4. ShellSort.java

**Code:**

```

import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;

public class ShellSort {
    public static void shellSort(int[] arr) {
        int n = arr.length;
        int pass = 1;
        for (int gap = n / 2; gap > 0; gap /= 2) {
            for (int i = gap; i < n; i++) {
                int temp = arr[i];
                int j = i;
                while (j >= gap && arr[j - gap] > temp) {
                    arr[j] = arr[j - gap];
                    j -= gap;
                }
                arr[j] = temp;
            }
            System.out.print("\nGap=" + gap + "\nPass" + pass + " : ");
            for (int i = 0; i < n; i++) {
                System.out.print(" " + arr[i]);
            }
            pass++;
        }
    }

    public static void main(String[] args) {

```

```

    int size = 5;
    int[] myArray = new int[size];

    try {
        File file = new File("C:\\Users\\Bhuravane\\eclipse-
workspace\\ADS\\src\\num.txt");
        Scanner scanner = new Scanner(file);

        for (int i = 0; i < size; i++) {
            if (scanner.hasNextInt()) {
                myArray[i] = scanner.nextInt();
            }
        }
        scanner.close();
    } catch (FileNotFoundException e) {
        System.err.println("Failed to open file for reading");
        e.printStackTrace();
        return;
    }

    System.out.print("\n\nUnsorted Array: ");
    for (int i = 0; i < size; i++)
        System.out.print(myArray[i] + " ");

    shellSort(myArray);

    System.out.print("\n\nSorted Array: ");
    for (int i = 0; i < size; i++)
        System.out.print(myArray[i] + " ");
}
}

```

### Num.txt

1 5 6 9 2

### Output:

```

Unsorted Array: 1 5 6 9 2
Gap=2
Pass1 :  1 5 2 9 6|
Gap=1
Pass2 :  1 2 5 6 9

Sorted Array: 1 2 5 6 9

```

## 5. Linerar Search

### Code:

```
import java.util.Scanner;

public class LinerarSearch {

    public static int linearSearch(int[] num, int key)
    {
        for (int i = 0; i < num.length; i++)
        {
            if (num[i]==key)
            {
                return i;
            }
        }
        return -1;
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the size of array: ");
        int size=scanner.nextInt();
        int[] num = new int[size];
        System.out.print("Enter the elements: ");
        for (int i = 0; i < size; i++) {
            num[i] = scanner.nextInt();
        }
        System.out.print("Enter an element to search: ");
        int key = scanner.nextInt();
```

```

int result = linearSearch(num, key);
if (result != -1) {
    System.out.println("Element found at index: " + result);
} else {
    System.out.println("Element not found in array");
}
scanner.close();
}
}

```

### Output:

```

Enter the size of array: 5
Enter the elements: 1 9 3 7 6
Enter an element to search: 7
Element found at index: 3

```

## 6. Binary Search

### Code:

```

import java.util.Arrays;
import java.util.Scanner;

public class BinarySearchExample {

    public static int binarySearch(int[] num, int size, int key) {

        int low = 0, high = size - 1;

        while (low <= high) {

            int mid = (low + high) / 2;

            if (num[mid] == key) return mid;

            if (num[mid] < key) low = mid + 1;

            else high = mid - 1;

```

```

    }

    return -1;
}

public static void main(String[] args) {

    Scanner sc = new Scanner(System.in);

    System.out.print("Enter the size of the array: ");

    int size = sc.nextInt();

    int[] num = new int[size];

    System.out.println("Enter the elements: ");

    for (int i = 0; i < size; i++) num[i] = sc.nextInt();

    Arrays.sort(num);

    System.out.println("Sorted array: " + Arrays.toString(num));

    System.out.print("Enter the element to search: ");

    int key = sc.nextInt();

    int result = binarySearch(num, size, key);

    if (result != -1) System.out.println("Element found at index: " + result);

    else System.out.println("Element not found in the array");

    sc.close();

}

}

```

### Output:

```

Enter the size of the array: 5|
Enter the elements:
4
8
3
9
2
Sorted array: [2, 3, 4, 8, 9]
Enter the element to search: 8
Element found at index: 3

```