

18-11-2024

DSA CODING PRACTICE

1.Bubble sort

```
import java.util.*;
```

```
public class problem1 {  
    public static void bubbleSort(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]) {  
                    int temp = arr[j];  
                    arr[j] = arr[j + 1];  
                    arr[j + 1] = temp;  
                }  
            }  
        }  
    }  
}  
  
public static void main(String[] args) {  
    Scanner scanner = new Scanner(System.in);  
    System.out.print("Enter the number of elements in the array: ");  
    int n = scanner.nextInt();  
    int[] arr = new int[n];  
    System.out.println("Enter the elements of the array:");  
    for (int i = 0; i < n; i++) {  
        arr[i] = scanner.nextInt();  
    }  
    bubbleSort(arr);  
    System.out.println("Sorted array: " + Arrays.toString(arr));  
}  
}
```

```

C:\Users\shali\OneDrive\Desktop\DSA-CODING-PRACTICE\day-5>javac problem1.java

C:\Users\shali\OneDrive\Desktop\DSA-CODING-PRACTICE\day-5>java problem1
Enter the number of elements in the array: 5
Enter the elements of the array:
4 1 3 9 7
Sorted array: [1, 3, 4, 7, 9]

C:\Users\shali\OneDrive\Desktop\DSA-CODING-PRACTICE\day-5>java problem1
Enter the number of elements in the array: 10
Enter the elements of the array:
10 9 8 7 6 5 4 3 2 1
Sorted array: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

```

2.Quick sort

```
import java.util.*;
```

```

public class problem2 {

    public static void quickSort(int[] arr, int low, int high) {

        if (low < high) {

            int pi = partition(arr, low, high);

            quickSort(arr, low, pi - 1);

            quickSort(arr, pi + 1, high);

        }

    }
}

```

```

    public static int partition(int[] arr, int low, int high) {

        int pivot = arr[high];

        int i = low - 1;

        for (int j = low; j < high; j++) {

            if (arr[j] <= pivot) {

                i++;

                int temp = arr[i];

                arr[i] = arr[j];

                arr[j] = temp;

            }

        }

        int temp = arr[i + 1];

        arr[i + 1] = arr[high];
    }
}

```

```

        arr[high] = temp;

        return i + 1;
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the number of elements in the array: ");

        int n = scanner.nextInt();

        int[] arr = new int[n];

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

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

        quickSort(arr, 0, n - 1);

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

```

```

C:\Users\shali\OneDrive\Desktop\DSA-CODING-PRACTICE\day-5>javac problem2.java

C:\Users\shali\OneDrive\Desktop\DSA-CODING-PRACTICE\day-5>java problem2
Enter the number of elements in the array: 5
Enter the elements of the array:
4 1 3 9 5
Sorted array: [1, 3, 4, 5, 9]

C:\Users\shali\OneDrive\Desktop\DSA-CODING-PRACTICE\day-5>java problem2
Enter the number of elements in the array: 9
Enter the elements of the array:
2 1 6 10 4 1 3 9 7
Sorted array: [1, 1, 2, 3, 4, 6, 7, 9, 10]

```

3.Non Repeating Character

```

import java.util.*;

public class problem3 {

    public static char firstNonRepeatingCharacter(String s) {
        Map<Character, Integer> frequencyMap = new LinkedHashMap<>();

        for (char c : s.toCharArray()) {
            frequencyMap.put(c, frequencyMap.getOrDefault(c, 0) + 1);
        }
    }
}

```

```

    }
    for (Map.Entry<Character, Integer> entry : frequencyMap.entrySet()) {
        if (entry.getValue() == 1) {
            return entry.getKey();
        }
    }
    return '$';
}

```

```

public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    System.out.print("Enter the string: ");
    String s = scanner.nextLine();
    char result = firstNonRepeatingCharacter(s);
    System.out.println(result == '$' ? -1 : result);
}
}

```

```

C:\Users\shali\OneDrive\Desktop\DSA-CODING-PRACTICE\day-5>javac problem3.java
C:\Users\shali\OneDrive\Desktop\DSA-CODING-PRACTICE\day-5>java problem3
Enter the string: racecar
e
C:\Users\shali\OneDrive\Desktop\DSA-CODING-PRACTICE\day-5>java problem3
Enter the string: aaabbbccc
$

```

4.Edit Distance

```
import java.util.*;
```

```

public class problem4 {
    public static int minOperations(String s1, String s2) {
        int m = s1.length(), n = s2.length();
        int[][] dp = new int[m + 1][n + 1];

        for (int i = 0; i <= m; i++) {
            for (int j = 0; j <= n; j++) {

```

```

        if (i == 0) {
            dp[i][j] = j;
        } else if (j == 0) {
            dp[i][j] = i;
        } else if (s1.charAt(i - 1) == s2.charAt(j - 1)) {
            dp[i][j] = dp[i - 1][j - 1];
        } else {
            dp[i][j] = 1 + Math.min(dp[i - 1][j - 1], Math.min(dp[i - 1][j], dp[i][j - 1]));
        }
    }
}

return dp[m][n];
}

```

```

public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);

    System.out.print("Enter the first string: ");
    String s1 = scanner.nextLine();

    System.out.print("Enter the second string: ");
    String s2 = scanner.nextLine();

    System.out.println("Minimum operations required: " + minOperations(s1, s2));
}
}

```

```

C:\Users\shali\OneDrive\Desktop\DSA-CODING-PRACTICE\day-5>javac problem4.java

C:\Users\shali\OneDrive\Desktop\DSA-CODING-PRACTICE\day-5>java problem4
Enter the first string: geek
Enter the second string: gesek
Minimum operations required: 1

C:\Users\shali\OneDrive\Desktop\DSA-CODING-PRACTICE\day-5>java problem4
Enter the first string: gfg
Enter the second string: gfg
Minimum operations required: 0

```

5.K-largest element

```
import java.util.*;

public class problem5 {

    public static List<Integer> kLargestElements(int[] arr, int k) {
        PriorityQueue<Integer> minHeap = new PriorityQueue<>();
        for (int num : arr) {
            minHeap.add(num);
            if (minHeap.size() > k) {
                minHeap.poll();
            }
        }
        List<Integer> result = new ArrayList<>(minHeap);
        result.sort(Collections.reverseOrder());
        return result;
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of elements in the array: ");
        int n = scanner.nextInt();
        int[] arr = new int[n];
        System.out.println("Enter the elements of the array:");
        for (int i = 0; i < n; i++) {
            arr[i] = scanner.nextInt();
        }
        System.out.print("Enter the value of k: ");
        int k = scanner.nextInt();
        List<Integer> result = kLargestElements(arr, k);
        System.out.println("K largest elements in decreasing order: " + result);
    }
}
```

```

C:\Users\shali\OneDrive\Desktop\DSA-CODING-PRACTICE\day-5>javac problem5.java

C:\Users\shali\OneDrive\Desktop\DSA-CODING-PRACTICE\day-5>java problem5
Enter the number of elements in the array: 5
Enter the elements of the array:
12 5 787 1 23
Enter the value of k: 2
K largest elements in decreasing order: [787, 23]

C:\Users\shali\OneDrive\Desktop\DSA-CODING-PRACTICE\day-5>java problem5
Enter the number of elements in the array: 7
Enter the elements of the array:
1 23 12 9 30 2 50
Enter the value of k: 3
K largest elements in decreasing order: [50, 30, 23]

```

6. Form the largest number

```
import java.util.*;
```

```

public class problem6 {

    public static String largestNumber(int[] arr) {

        String[] strArr = Arrays.stream(arr)
                                .mapToObj(String::valueOf)
                                .toArray(String[]::new);

        Arrays.sort(strArr, (a, b) -> (b + a).compareTo(a + b));

        if (strArr[0].equals("0")) return "0";

        return String.join("", strArr);

    }

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the number of elements in the array: ");

        int n = scanner.nextInt();

        int[] arr = new int[n];

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

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

            arr[i] = scanner.nextInt();

        }

        System.out.println("Largest number formed: " + largestNumber(arr));

    }
}

```

}

```
C:\Users\shali\OneDrive\Desktop\DSA-CODING-PRACTICE\day-5>javac problem6.java
```

```
C:\Users\shali\OneDrive\Desktop\DSA-CODING-PRACTICE\day-5>java problem6
```

```
Enter the number of elements in the array: 5
```

```
Enter the elements of the array:
```

```
3 30 34 5 9
```

```
Largest number formed: 9534330
```

```
C:\Users\shali\OneDrive\Desktop\DSA-CODING-PRACTICE\day-5>java problem6
```

```
Enter the number of elements in the array: 4
```

```
Enter the elements of the array:
```

```
54 546 548 60
```

```
Largest number formed: 6054854654
```