

DSA Coding Practice-6

1. Bubble Sort:

The screenshot displays a coding practice interface for the Bubble Sort problem. The left sidebar shows the problem status: "Problem Solved Successfully" with a green checkmark. It lists "Test Cases Passed: 1115 / 1115", "Attempts: Correct / Total: 1 / 1", "Accuracy: 100%", "Points Scored: 2 / 2", and "Time Taken: 0.66". The right sidebar shows the Java code for the bubbleSort function, which is a public static void method that iterates through the array and swaps adjacent elements if they are in the wrong order.

```
1 // Driver Code Ends
9 // User function Template for Java
10
11 class Solution {
12     public static void bubbleSort(int arr[]) {
13         int n = arr.length;
14         for (int i = 0; i < n - 1; i++) {
15             boolean swapped = false;
16             for (int j = 0; j < n - i - 1; j++) {
17                 if (arr[j] > arr[j + 1]) {
18                     int temp = arr[j];
19                     arr[j] = arr[j + 1];
20                     arr[j + 1] = temp;
21                     swapped = true;
22                 }
23             }
24             if (!swapped) break;
25         }
26     }
27 }
28 // Driver Code Ends
```

2.Quick Sort:

The screenshot displays a coding practice interface for the Quick Sort problem. The left sidebar shows the problem status: "Problem Solved Successfully" with a green checkmark. It lists "Test Cases Passed: 1120 / 1120", "Attempts: Correct / Total: 1 / 1", "Accuracy: 100%", "Points Scored: 4 / 4", and "Time Taken: 0.59". The right sidebar shows the Java code for the quickSort and partition functions. The quickSort function is a static void method that recursively sorts the array, and the partition function is a static int method that returns the pivot index.

```
1 // Driver Code Ends
30
31
32 class Solution {
33     static void quickSort(int arr[], int low, int high) {
34         if (low < high) {
35             int pivotIndex = partition(arr, low, high);
36             quickSort(arr, low, pivotIndex - 1);
37             quickSort(arr, pivotIndex + 1, high);
38         }
39     }
40     static int partition(int arr[], int low, int high) {
41         int pivot = arr[high];
42         int i = low - 1;
43         for (int j = low; j < high; j++) {
44             if (arr[j] <= pivot) {
45                 i++;
46                 int temp = arr[i];
47                 arr[i] = arr[j];
48                 arr[j] = temp;
49             }
50         }
51         int temp = arr[i + 1];
52         arr[i + 1] = arr[high];
53         arr[high] = temp;
54         return i + 1;
55     }
56 }
```

3.Non Repeating Character:

Problem

Editorial

Submissions

Comments

Output Window

Compilation Results

Custom Input

Y.O.G.I. (AI Bot)

Problem Solved Successfully

Test Cases Passed

1130 / 1130

Attempts: Correct / Total

1 / 2

Accuracy: 50%

Points Scored

2 / 2

Your Total Score: 36

Time Taken

0.73

Solve Next

Java (1.8)

Average Time: 30m

Start Timer

```
1 // Driver Code Ends
29
30
31 // User function Template for Java
32
33 class Solution {
34     static char nonRepeatingChar(String s) {
35         Map<Character, Integer> frequencyMap = new HashMap<>();
36         for (char c : s.toCharArray()) {
37             frequencyMap.put(c, frequencyMap.getOrDefault(c, 0) + 1);
38         }
39         for (char c : s.toCharArray()) {
40             if (frequencyMap.get(c) == 1) {
41                 return c;
42             }
43         }
44         return '$';
45     }
46 }
47
48
```