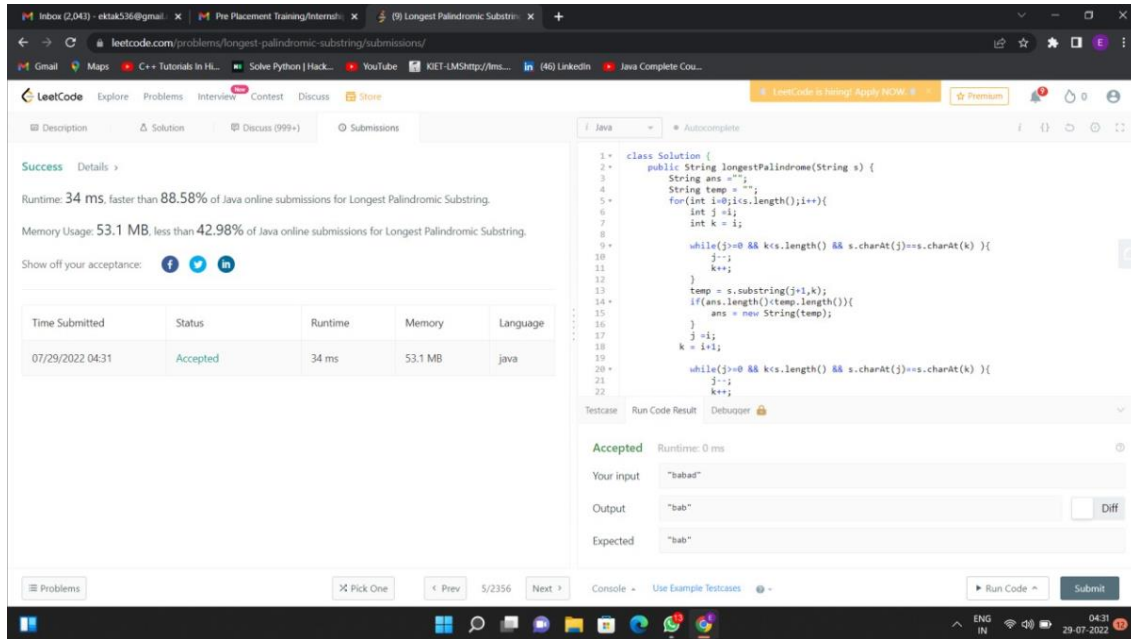


Akanksha Yadav

1900290120008

WEEK -4

PROBLEM-1(LEETCODE)



The screenshot shows a LeetCode submission for the problem "Longest Palindromic Substring". The submission is successful, with a runtime of 34 ms, faster than 88.58% of Java online submissions, and a memory usage of 53.1 MB, less than 42.98% of Java online submissions. The code is in Java and uses a brute-force approach to find the longest palindromic substring.

```
1 class Solution {
2     public String longestPalindrome(String s) {
3         String ans = "";
4         String temp = "";
5         for(int i=0; i<s.length(); i++){
6             int j = i;
7             int k = i;
8
9             while(j>0 && k<s.length() && s.charAt(j)==s.charAt(k)){
10                 j--;
11                 k++;
12             }
13             temp = s.substring(j+1,k);
14             if(ans.length()<temp.length()){
15                 ans = new String(temp);
16             }
17             j = i;
18             k = i+1;
19
20             while(j>0 && k<s.length() && s.charAt(j)==s.charAt(k)){
21                 j--;
22                 k++;
23             }
24         }
25         return ans;
26     }
27 }
```

Testcase: Run Code Result: Debugger

Accepted Runtime: 0 ms

Your input: "babad"

Output: "bab"

Expected: "bab"

Diff

PROBLEM-2

Intbox (2,043) · vktak53@gmail... · Pre Placement Training/Internat... · Jump Game - LeetCode · Longest Palindromic Substrin... · +

← → ↻ leetcode.com/problems/jump-game/submissions/

Gmail Maps C++ Tutorials in HL... Solve Python | Hack... YouTube KIET-UMGithub/Ins... (46) LinkedIn Java Complete Cou...

LeetCode Explore Problems Interview Contest Discuss Store

LearnCode is hiring! Apply NOW. 6 Premium




Description Solution Discuss (999+) Submissions

Java Autocomplete

Success Details >

Runtime: 3 ms, faster than 66.18% of Java online submissions for Jump Game.

Memory Usage: 68.1 MB, less than 44.52% of Java online submissions for Jump Game.

Show off your acceptance:   

Time Submitted	Status	Runtime	Memory	Language
07/29/2022 04:35	Accepted	3 ms	68.1 MB	java

```
1 class Solution {
2 public boolean canJump(int[] arr) {
3 if(arr.length == 1)
4 return true;
5 if(arr[0]==0)
6 return false;
7 int limit= arr[0], cJ = arr[0]; // 'c' an 'j'ump is the number we can jump
8 // limit is the index upto which we can jump as of now
9 for(int i=0;i<arr.length-1 && i<limit;i++){
10 cJ = Math.max(cJ,arr[i]); //checking if ability to jump(cJ) is increased
11 limit = i+cJ; // "i" is the current block and cJ is power to jump
12
13 if(i+arr[i] >= arr.length-1)
14 return true;
15 cJ-=1; //since we jumped 1 box our ability to jump reduces by 1
16 }
17 return false;
18 }
19 }
```

Testcase Run Code Result Debugger


Accepted Runtime: 0 ms

Your input [2,3,1,1,4]

Output true Diff

Expected true

Problems Pick One < Prev 55/2356 Next > Console Use Example Testcases Run Code Submit



ENG IN 04:35 29-07-2022