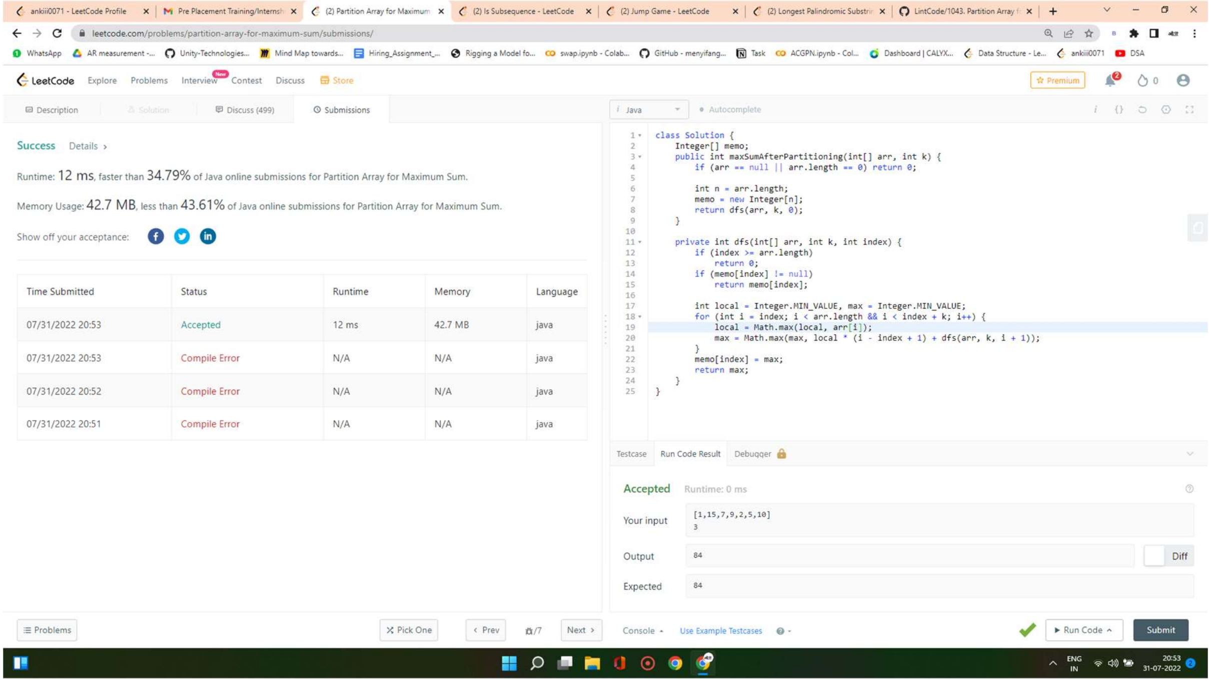
Leetcode submission



Success Details ›

Runtime: 12 ms, faster than 34.79% of Java online submissions for Partition Array for Maximum Sum.

Memory Usage: 42.7 MB, less than 43.61% of Java online submissions for Partition Array for Maximum Sum.

Show off your acceptance:

Time Submitted	Status	Runtime	Memory	Language
07/31/2022 20:53	Accepted	12 ms	42.7 MB	java
07/31/2022 20:53	Compile Error	N/A	N/A	java
07/31/2022 20:52	Compile Error	N/A	N/A	java
07/31/2022 20:51	Compile Error	N/A	N/A	java

```java
class Solution {
    Integer[] memo;
    public int maxSumAfterPartitioning(int[] arr, int k) {
        if (arr == null || arr.length == 0) return 0;

        int n = arr.length;
        memo = new Integer[n];
        return dfs(arr, k, 0);
    }

    private int dfs(int[] arr, int k, int index) {
        if (index >= arr.length)
            return 0;
        if (memo[index] != null)
            return memo[index];

        int local = Integer.MIN_VALUE, max = Integer.MIN_VALUE;
        for (int i = index; i < arr.length && i < index + k; i++) {
            local = Math.max(local, arr[i]);
            max = Math.max(max, local * (i - index + 1) + dfs(arr, k, i + 1));
        }
        memo[index] = max;
        return max;
    }
}
```

Accepted Runtime: 0 ms

Your input

```
[1,15,7,9,2,5,10]
3
```

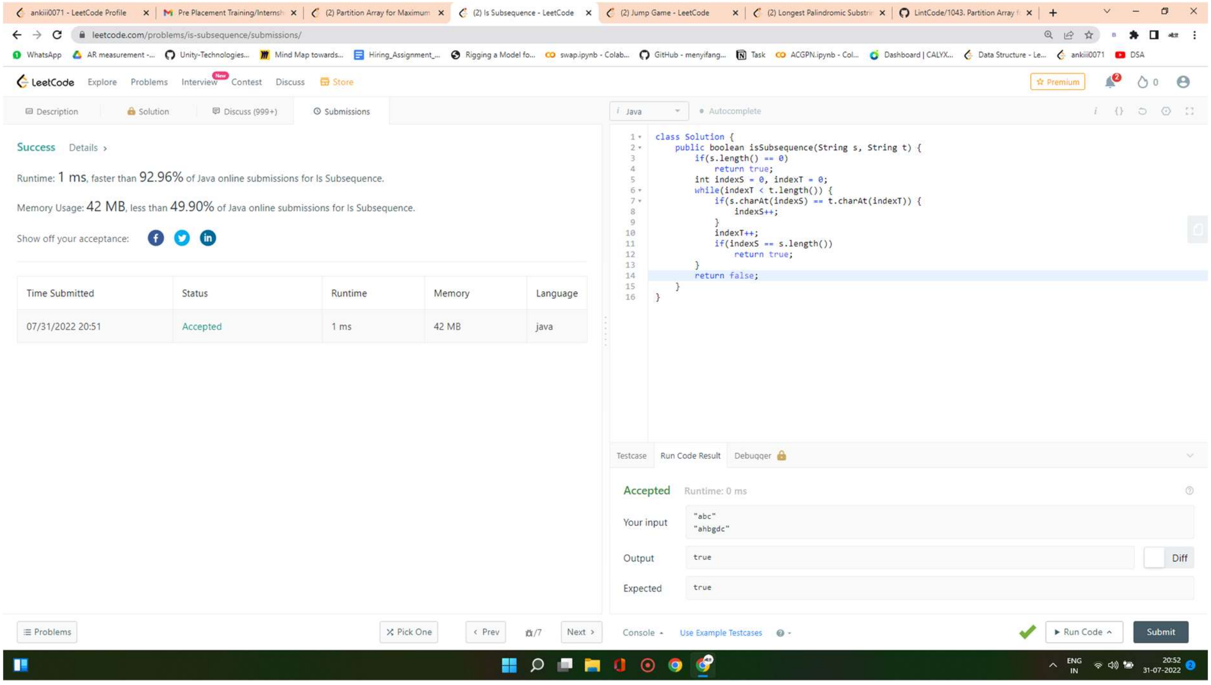
Output

```
84
```

Expected

```
84
```

20:53 31-07-2022



Success Details ›

Runtime: 1 ms, faster than 92.96% of Java online submissions for Is Subsequence.

Memory Usage: 42 MB, less than 49.90% of Java online submissions for Is Subsequence.

Show off your acceptance:

Time Submitted	Status	Runtime	Memory	Language
07/31/2022 20:51	Accepted	1 ms	42 MB	java

```java
class Solution {
    public boolean isSubsequence(String s, String t) {
        if(s.length() == 0)
            return true;
        int indexS = 0, indexT = 0;
        while(indexT < t.length()) {
            if(s.charAt(indexS) == t.charAt(indexT)) {
                indexS++;
            }
            indexT++;
            if(indexS == s.length())
                return true;
        }
        return false;
    }
}
```

Accepted Runtime: 0 ms

Your input

```
"abc"
"ahbgdc"
```

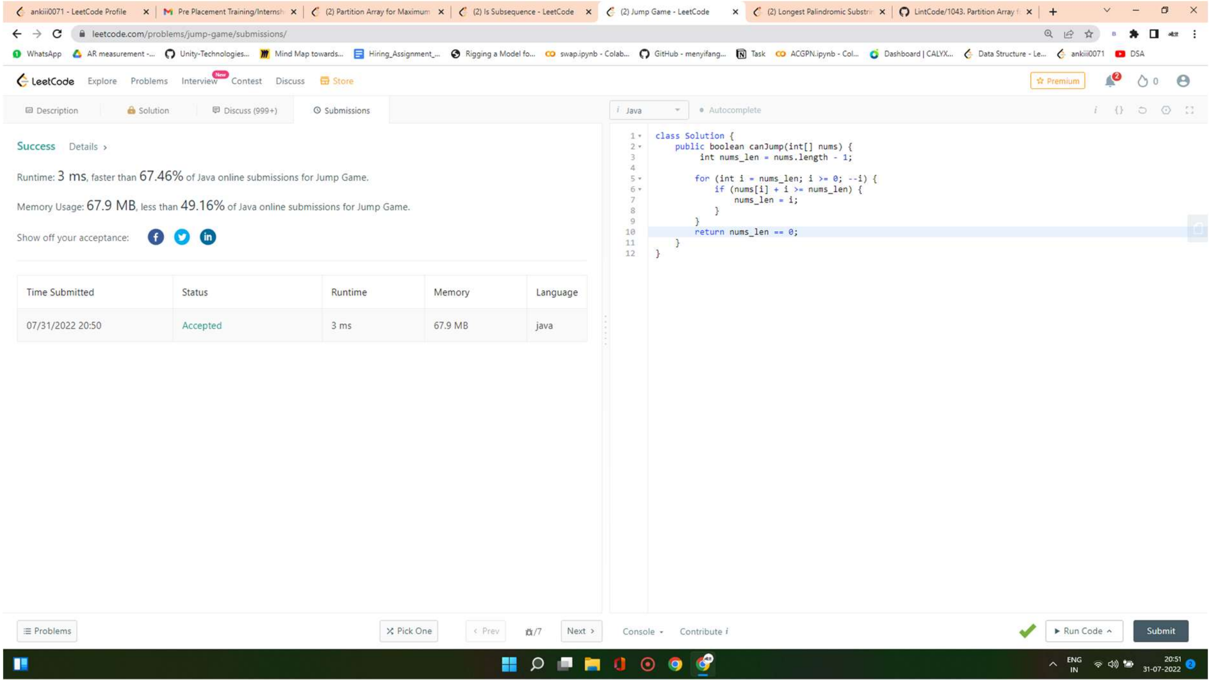
Output

```
true
```

Expected

```
true
```

20:52 31-07-2022



Success Details ›

Runtime: 3 ms, faster than 67.46% of Java online submissions for Jump Game.

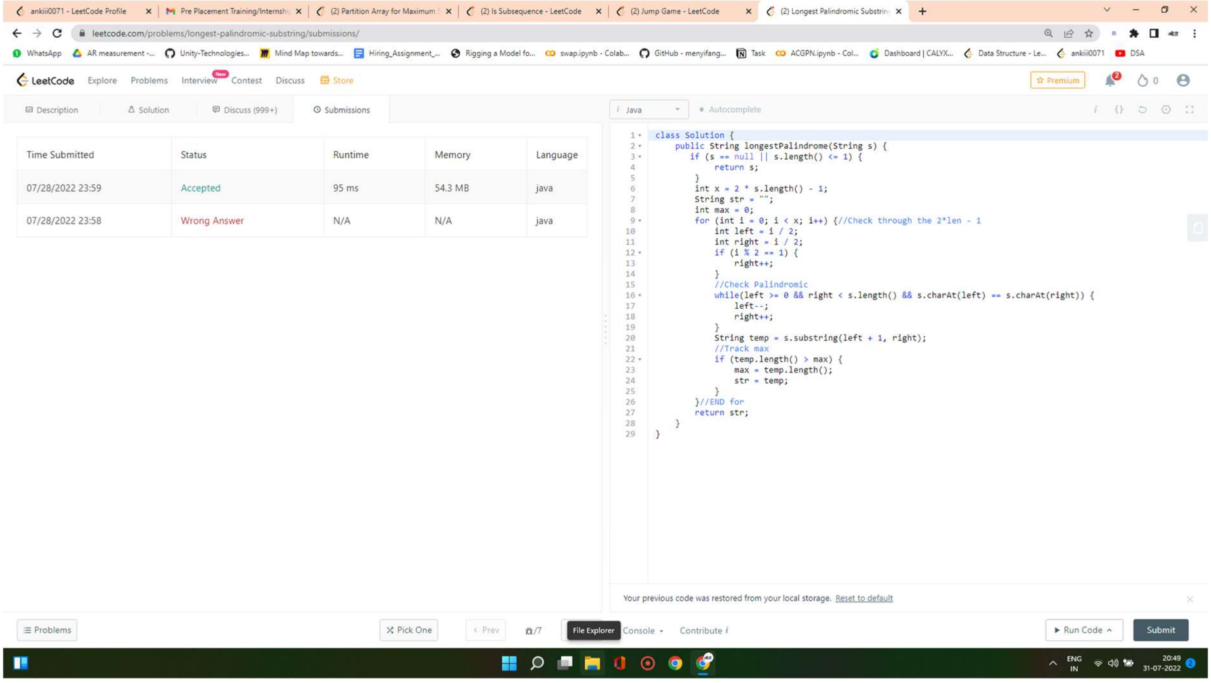
Memory Usage: 67.9 MB, less than 49.16% of Java online submissions for Jump Game.

Show off your acceptance:

Time Submitted	Status	Runtime	Memory	Language
07/31/2022 20:50	Accepted	3 ms	67.9 MB	java

```java
class Solution {
    public boolean canJump(int[] nums) {
         int nums_len = nums.length - 1;

        for (int i = nums_len; i >= 0; --i) {
            if (nums[i] + i >= nums_len) {
                nums_len = i;
            }
        }
        return nums_len == 0;
    }
}
```



Time Submitted	Status	Runtime	Memory	Language
07/28/2022 23:59	Accepted	95 ms	54.3 MB	java
07/28/2022 23:58	Wrong Answer	N/A	N/A	java

```java
class Solution {
    public String longestPalindrome(String s) {
       if (s == null || s.length() <= 1) {
           return s;
        }
        int x = 2 * s.length() - 1;
        String str = "";
        int max = 0;
        for (int i = 0; i < x; i++) {//Check through the 2*len - 1
            int left = i / 2;
            int right = i / 2;
            if (i % 2 == 1) {
                right++;
            }
            //Check Palindromic
            while(left >= 0 && right < s.length() && s.charAt(left) == s.charAt(right)) {
                left--;
                right++;
            }
            String temp = s.substring(left + 1, right);
            //Track max
            if (temp.length() > max) {
                max = temp.length();
                str = temp;
            }
        }//END for
        return str;
    }
}
```

Your previous code was restored from your local storage. Reset to default