

Leetcode submission

LeetCode Submission: Partition Array for Maximum Sum

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

```
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;
    }
}
```

Testcase: Run Code Result: Debugger

Accepted Runtime: 0 ms

Your input: [1,15,7,9,2,5,10]
3

Output: 84

Expected: 84

LeetCode Submission: Is Subsequence

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

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

Testcase: Run Code Result: Debugger

Accepted Runtime: 0 ms

Your input: "abc"
"ahbgdc"

Output: true

Expected: true

ankii0071 - LeetCode Profile | Pre Placement Training/Internships | (2) Partition Array for Maximum Sum | (2) Is Subsequence - LeetCode | (2) Jump Game - LeetCode | (2) Longest Palindromic Substring | LeetCode/1043. Partition Array | +

leetcode.com/problems/jump-game/submissions/

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: [f](#) [t](#) [in](#)

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

```
1 class Solution {
2     public boolean canJump(int[] nums) {
3         int nums_len = nums.length - 1;
4         for (int i = nums_len; i >= 0; --i) {
5             if (nums[i] + 1 >= nums_len) {
6                 nums_len = i;
7             }
8         }
9         return nums_len == 0;
10    }
11 }
12 }
```

Console | Contribute i

Run Code Submit

ankii0071 - LeetCode Profile | Pre Placement Training/Internships | (2) Partition Array for Maximum Sum | (2) Is Subsequence - LeetCode | (2) Jump Game - LeetCode | (2) Longest Palindromic Substring | +

leetcode.com/problems/longest-palindromic-substring/submissions/

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

```
1 class Solution {
2     public String longestPalindrome(String s) {
3         if (s == null || s.length() <= 1) {
4             return s;
5         }
6         int x = 2 * s.length() - 1;
7         String str = "";
8         int max = 0;
9         for (int i = 0; i < x; i++) { //Check through the 2*len - 1
10             int left = i / 2;
11             int right = i / 2;
12             if (i % 2 == 1) {
13                 right++;
14             }
15             //Check Palindromic
16             while (left >= 0 && right < s.length() && s.charAt(left) == s.charAt(right)) {
17                 left--;
18                 right++;
19             }
20             String temp = s.substring(left + 1, right);
21             //Track max
22             if (temp.length() > max) {
23                 max = temp.length();
24                 str = temp;
25             }
26         } //END for
27         return str;
28     }
29 }
```

Your previous code was restored from your local storage. [Reset to default](#)

File Explorer | Console | Contribute i

Run Code Submit