

Cracking the Code: Data Structures and Algorithms (DSA) in JavaScript - II

Marks - 150

Note: Attempt this assignment after studying pre-recorded content and after attending Live lectures of this course.

Create your Leetcode Profiles and solve these questions there. Share your solved questions link along with the Time complexity and Space Complexity of your solution in a doc when you submit your assignment.

Submit the optimised solution for all the questions.

WARNING !! Don't try to copy from somewhere else. We can call any student randomly to explain their solutions, and if we find a discrepancy, you will be provided with 0.

Example Question with Solution

(Please note that you need to submit your answers in exact same format, including the description)

For Example :

Question <https://leetcode.com/problems/concatenation-of-array/description/>

Solution Link -

<https://leetcode.com/problems/concatenation-of-array/submissions/1257621340/>

Add screenshot:

The screenshot shows a LeetCode submission page for a problem. The top navigation bar includes 'Problem List', 'Description', 'Accepted' (highlighted), 'Editorial', 'Solutions', and 'Submissions'. The submission was made by user 'b72Ua3YCR' at May 14, 2024 13:06. The status is 'Accepted'. The runtime is 66 ms (Beats 54.72% of users with JavaScript) and the memory usage is 51.58 MB (Beats 76.26% of users with JavaScript). The code editor displays the following JavaScript code:

```
1 /**
2  * @param {number[]} nums
3  * @return {number[]}
4  */
5 var getConcatenation = function(nums)
6 {
7     const len = nums.length
8     for (let i = 0; i < len; i++) {
9         nums.push(nums[i])
10    }
11 }
12 
```

The code is saved at Ln 9, Col 16. Below the code editor is a 'Testcase' section with 'Case 1' and 'Case 2' buttons, and a text input field containing 'nums = [1,2,1]'. Buttons for 'Run' and 'Submit' are also present.

Description:

Time Complexity: O(n)

Iterating to the entire n-sized array and for each array element performing O(1) operation of push.

Space complexity: O(1)

No Extra space required for the Program Execution

**Note: Write about the approach and algorithm used in each solution in detail in the description section.
A short description may lead to a reduction in marks.**

- Given the head of a singly linked list, return true if it is a palindrome or false otherwise [Leetcode 234](#) **(20 marks)**
- Given the head of a linked list, remove the nth node from the end of the list and return its head. [Leetcode 19](#) **(20 marks)**
- Given a string s containing just the characters '(', ')', '{', '}', '[', and ']', determine if the input string is valid.
[Valid Parentheses](#) **(20 marks)**
- The next greater element of some element x in an array is the first greater element that is to the right of x in the same array.
[Next Greater Element](#) **(20 marks)**
- Given a binary tree, find its minimum depth. The minimum depth is the number of nodes along the shortest path from the root node down to the nearest leaf node. [Minimum Depth of Binary Tree](#) **(25 marks)**
- Given the root of a binary tree, return the postorder traversal of its nodes' values.
[Binary Tree Postorder Traversal](#) **(20 marks)**
- There is a singly-linked list head, and you have to delete a node in it. [Leetcode 237](#) **(25 marks)**