Data Structure Algorithm Assignment-2

Marks - 100

Note: Attempt this assignment after studying pre-recorded content and after attending live lectures of DSA-1

1. Given an array of integer ‘nums’ and an integer ‘target’, return indices of the two numbers such that they add up to the ‘target’. Leetcode 1 **(10 marks)**

2. Given an integer array nums, return all the triplets [nums[i], nums[j], nums[k]] such that i != j, i != k, and j != k, and nums[i] + nums[j] + nums[k] == 0. Leetcode 15 **(15 marks)**

3. Your friend is typing his ‘name’ into a keyboard. Sometimes, when typing a character ‘c’, the key might get long pressed, and the character will be ‘typed’ 1 or more times.
   You examine the typed characters of the keyboard. Return ‘True’ if it is possible that it was your friend's name, with some characters (possibly none) being long pressed.
   Leetcode 925 **(10 marks)**

4. You are given an integer array arr of length n that represents a permutation of the integers in the range [0, n - 1].
   We split arr into some number of chunks (i.e., partitions), and individually sort each chunk. After concatenating them, the result should equal the sorted array.
   Return the largest number of chunks we can make to sort the array
   Leetcode 769 **(15 marks)**

5. Given an array nums with n objects colored red, white, or blue, sort them in place so that objects of the same color are adjacent, with the colors in the order red, white, and blue.
   We will use the integers 0, 1, and 2 to represent the colors red, white, and blue, respectively.
   You must solve this problem without using the library's sort function.
   Leetcode 75 **(20 marks)**

6. Given an integer array nums, find the subarray with the largest sum, and return its sum.
   Leetcode 53 **(15 marks)**

7. Given an integer array nums, return an array answer such that answer[i] is equal to the product of all the elements of nums except nums[i].
   The product of any prefix or suffix of nums is guaranteed to fit in a 32-bit integer.
   You must write an algorithm that runs in O(n) time and without using the division operation.
   Leetcode 238 **(15 marks)**

Guidelines for submission: Create a LeetCode profile and solve the specified questions there. Share the links to your solved questions with their time and space complexities along with a description of explanations. Additionally, include screenshots of your solutions in a document when you submit your assignment. Also, try to submit the optimised solution for all the questions.

Example:

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

Solution Link -
https://leetcode.com/problems/concatenation-of-array/submissions/1257621340/

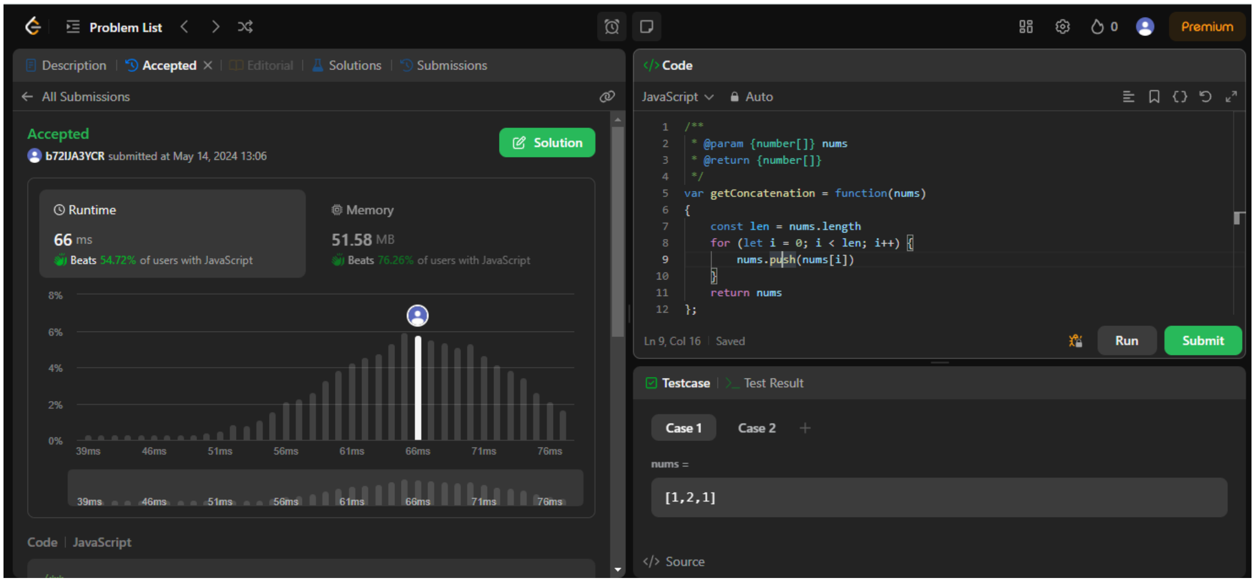
Description:
Time Complexity: O(n)

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 is required for the Program Execution

Screenshot:



Accepted

b72IJA3YCR submitted at May 14, 2024 13:06

Runtime
66 ms
Beats 54.72% of users with JavaScript

Memory
51.58 MB
Beats 76.26% of users with JavaScript

```
/**
 * @param {number[]} nums
 * @return {number[]}
 */
var getConcatenation = function(nums)
{
    const len = nums.length
    for (let i = 0; i < len; i++) {
        nums.push(nums[i])
    }
    return nums
};
```

Ln 9, Col 16 | Saved

nums =

[1,2,1]