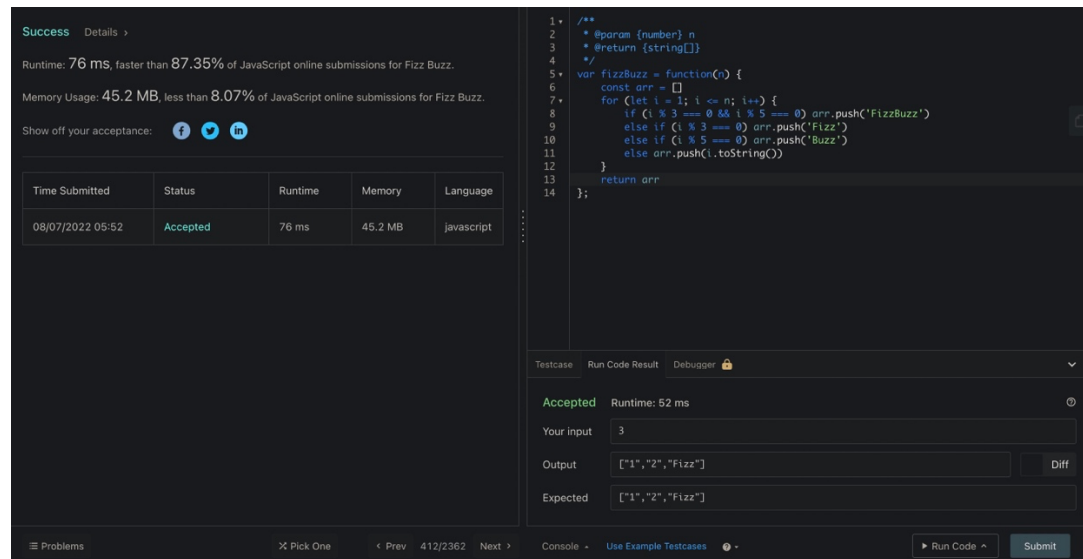


Nama : Raihan Romzi R

Batch : 9

1. FizzBuzz

a. Status Accepted



The screenshot shows a code execution interface with a dark theme. On the left, a 'Success' message indicates the code passed all tests. It provides performance metrics: 'Runtime: 76 ms, faster than 87.35% of JavaScript online submissions for Fizz Buzz.' and 'Memory Usage: 45.2 MB, less than 8.07% of JavaScript online submissions for Fizz Buzz.' Below this is a table with submission details.

Time Submitted	Status	Runtime	Memory	Language
08/07/2022 05:52	Accepted	76 ms	45.2 MB	javascript

On the right, the JavaScript code for the FizzBuzz function is displayed. It uses a for loop to iterate from 1 to n, pushing 'FizzBuzz', 'Fizz', 'Buzz', or the number itself into an array based on divisibility by 3 and 5.

```
1 /**  
2  * @param {number} n  
3  * @return {string[]}  
4  */  
5  var fizzBuzz = function(n) {  
6    const arr = []  
7    for (let i = 1; i <= n; i++) {  
8      if (i % 3 === 0 && i % 5 === 0) arr.push('FizzBuzz')  
9      else if (i % 3 === 0) arr.push('Fizz')  
10     else if (i % 5 === 0) arr.push('Buzz')  
11     else arr.push(i.toString())  
12   }  
13   return arr  
14 };
```

At the bottom right, a 'Testcase' tab shows the execution result. It indicates 'Accepted' with a runtime of 52 ms. The input is '3', and the output is '["1","2","Fizz"]', which matches the expected output.

b. Code

```
const fizzBuzz = function (n) {  
  const arr = []  
  for (let i = 1; i <= n; i++) {  
    if (i % 3 === 0 && i % 5 === 0) arr.push('FizzBuzz')  
    else if (i % 3 === 0) arr.push('Fizz')  
    else if (i % 5 === 0) arr.push('Buzz')  
    else arr.push(i.toString())  
  }  
  return arr  
};
```

c. Explanation

Membuat sebuah array kosong, yang akan diisi data string FizzBuzz, Fizz, Buzz, atau angka jika memenuhi kondisi yang sudah ditentukan.




2. Missing Number

a. Status Accepted

Success Details >

Runtime: 125 ms, faster than 35.27% of JavaScript online submissions for Missing Number.

Memory Usage: 44.7 MB, less than 28.94% of JavaScript online submissions for Missing Number.

Show off your acceptance:   

Time Submitted	Status	Runtime	Memory	Language
08/07/2022 07:12	Accepted	125 ms	44.7 MB	javascript
08/07/2022 07:04	Wrong Answer	N/A	N/A	javascript
08/07/2022 07:02	Wrong Answer	N/A	N/A	javascript

```
1 *  
2 * @param {number[]} nums  
3 * @return {number}  
4 */  
5 var missingNumber = function(nums) {  
6     let arrayFinal = Array.from({length: nums.length + 1}, (v, i) => i);  
7     let numsArrSorted = nums.sort(function (a, b) {  
8         if (a > b) return 1;  
9         if (a < b) return -1;  
10        return 0;  
11    });  
12    for (let i = 0; i <= nums.length + 1; i++) {  
13        if (numsArrSorted[i] !== arrayFinal[i]) {  
14            return arrayFinal[i]  
15        }  
16    }  
17    };  
18 }
```

Testcase Run Code Result Debugger

Accepted Runtime: 64 ms

Your Input [3,0,1]

Output 2 Diff

Expected 2

b. Code

```
let missing_number = function (nums) {  
    let arrayFinal = Array.from({length: nums.length + 1}, (v,  
i) => i);  
    let numsArrSorted = nums.sort(function (a, b) {  
        if (a > b) return 1;  
        if (a < b) return -1;  
        return 0;  
    })  
    for (let i = 0; i <= nums.length + 1; i++) {  
        if (numsArrSorted[i] !== arrayFinal[i]) {  
            return arrayFinal[i]  
        }  
    }  
};
```

c. Explanation

Membuat arrayFinal yang berisi angka dan berjumlah sebanyak panjang array inputan(nums) + 1. Selanjutnya mengurutkan array inputan (nums), yang akan digunakan untuk pembandingan, jika ditemukan ketidakcocokan maka angka tersebut yang tidak ada di array inputan (nums).

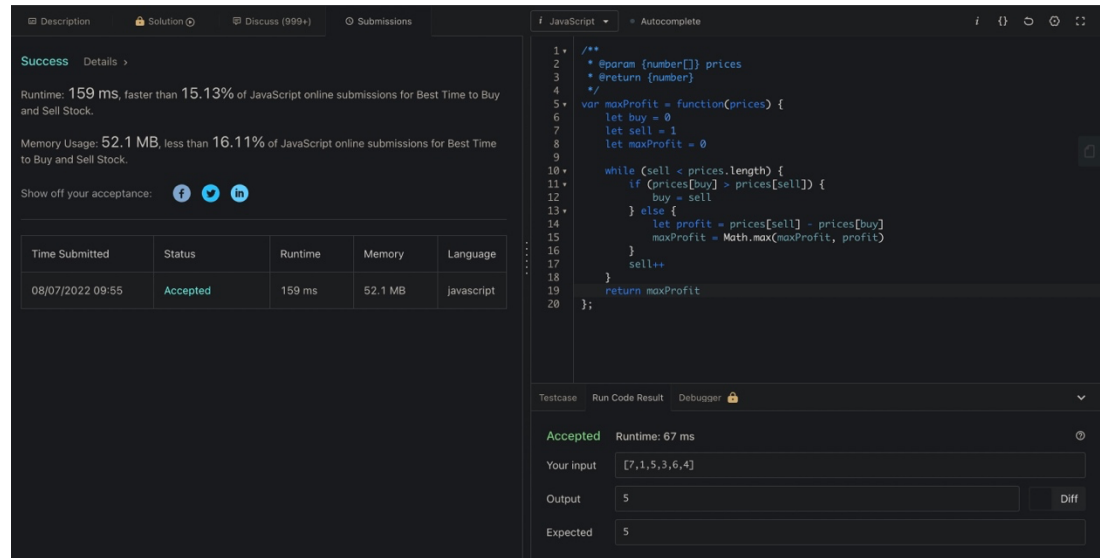
d. Reference

https://developer.mozilla.org/enUS/docs/Web/JavaScript/Reference/Global_Objects/Array/from

<https://www.javascripttutorial.net/javascript-array-sort/>

3. Best Time to Buy and Sell Stock

a. Status Accepted



The screenshot shows a LeetCode submission interface. On the left, a 'Success' message indicates the submission was accepted. Below this, a table lists submission details:

Time Submitted	Status	Runtime	Memory	Language
08/07/2022 09:55	Accepted	159 ms	52.1 MB	javascript

On the right, the JavaScript code for the solution is displayed. It uses a greedy algorithm to find the maximum profit by iterating through the price array and updating the buy and sell indices.

b. Code

```
let maxProfit = function (prices) {  
    let buy = 0  
    let sell = 1  
    let maxProfit = 0  
  
    while (sell < prices.length) {  
        if (prices[buy] > prices[sell]) {  
            buy = sell  
        } else {  
            let profit = prices[sell] - prices[buy]  
            maxProfit = Math.max(maxProfit, profit)  
        }  
        sell++  
    }  
    return maxProfit  
}
```

c. Explanation

Langkah pertama adalah kita harus loop ke seluruh elemen array, jika buy lebih besar daripada sell, maka index harus bergeser 1, jika tidak maka kita bisa mulai menghitung profit dengan mengurangi nilai sell dan nilai buy. Selanjutnya kita menghitung nilai profit maksimal.

d. Reference

<https://leetcode.com/problems/best-time-to-buy-and-sell-stock/discuss/1735550/Python-Javascript-Easy-solution-with-very-clear-Explanation>