

Fizz Buzz:

<https://leetcode.com/problems/fizz-buzz/description/?envType=featured-list&envId=top-interview-questions?envType=featured-list&envId=top-interview-questions>

1. Python

</> Code

Python ▾ 🔒 Auto

```
1 class Solution(object):
2     def fizzBuzz(self, n):
3         answer = []
4         for i in range(1, n+1):
5             if (i % 3 == 0 and i % 5 == 0):
6                 answer.append("FizzBuzz")
7             elif (i % 3 == 0):
8                 answer.append("Fizz")
9             elif (i % 5 == 0):
10                answer.append("Buzz")
11            else:
12                answer.append(str(i))
13        return answer
```

Ln 11, Col 18 | Saved to local

 Run Submit

✓ Testcase | >_ Test Result

Accepted Runtime: 25 ms

• Case 1 • Case 2 • Case 3

2. JavaScript

</> Code

JavaScript ▾ 🔒 Auto

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

Ln 13, Col 34 | Saved to local

 Run Submit

✓ Testcase | >_ Test Result

Accepted Runtime: 92 ms

• Case 1 • Case 2 • Case 3

Search Insert Position:

<https://leetcode.com/problems/search-insert-position/description/>

Python

Description | Editorial | Solutions | Submissions

← All Submissions

Accepted

Hadeer Khaled Elmalah submitted at Mar 03, 2024 01:03

Editorial

Solution

Runtime

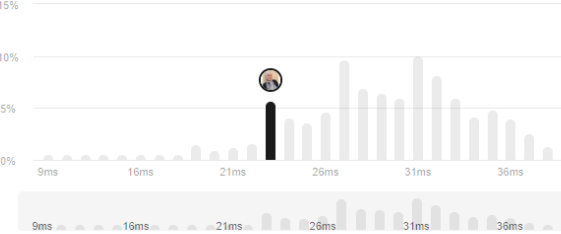
23 ms

Beats 93.60% of users with Python

Memory

12.48 MB

Beats 48.17% of users with Python



Code | Python

</> Code

Python Auto

```
1 class Solution(object):
2     def searchInsert(self, nums, target):
3         low = 0
4         high = len(nums) - 1
5         mid = 0
6         while low <= high:
7             mid = (high + low) // 2
8             if nums[mid] < target:
9                 low = mid + 1
10            elif nums[mid] > target:
11                high = mid - 1
12            else:
13                return mid
14        return low
15
16
```

Ln 10, Col 37 | Saved to local

Run Submit

Testcase Test Result

You have attempted to run code too soon. Please try again in a few seconds, or [Subscribe](#) to reduce wait time.

Move Zeroes:

<https://leetcode.com/problems/move-zeroes/description/?envType=featured-list&envId=top-interview-questions?envType=featured-list&envId=top-interview-questions>

1. Python

Python Auto

```
1 class Solution(object):
2     def moveZeroes(self, nums):
3         left = 0
4         for right in range(len(nums)):
5             if nums[right]:
6                 nums[left], nums[right] = nums[right], nums[left]
7                 left += 1
8         return nums
9
```

Ln 5, Col 26 | Saved to local Run Submit

☒ Testcase | Test Result

Accepted Runtime: 36 ms

• Case 1 • Case 2

2. Javascript

JavaScript Auto

```
1 var moveZeroes = function(nums) {
2     let left = 0;
3     for (let right = 0; right < nums.length; right++) {
4         if (nums[right] !== 0) {
5             nums[left] = nums[right];
6             left++;
7         }
8     }
9     while (left < nums.length) {
10         nums[left] = 0;
11         left++;
12     }
13     return nums;
14 };
```

Ln 2, Col 5 | Saved to local Run Submit

☒ Testcase | Test Result

Accepted Runtime: 82 ms

• Case 1 • Case 2

Climbing Stairs:

<https://leetcode.com/problems/climbing-stairs/description/?envType=featured-list&envId=top-interview-questions?envType=featured-list&envId=top-interview-questions>

Python ▾ Auto

2

class Solution(object):

3

def climbStairs(self, n):

4

if n == 1:

5

return 1

6

if n == 2:

7

return 2

8

arr = np.zeros(n + 1, dtype=int)

9

arr[0] = 1

10

arr[1] = 1

11

arr[2] = 2

12

for i in range(3, n + 1):

13

arr[i] = arr[i - 1] + arr[i - 2]

14

return arr[n]

Ln 14, Col 22 | Saved to local

Run

Submit

✓ Testcase | >_ Test Result

Accepted Runtime: 148 ms

• Case 1

• Case 2

Longest Substring Without Repeating Characters:

<https://leetcode.com/problems/longest-substring-without-repeating-characters/description/?envType=featured-list&envId=top-interview-questions?envType=featured-list&envId=top-interview-questions>

Python Auto

```
1 class Solution(object):
2     def lengthOfLongestSubstring(self, s):
3         dic = {}
4         count = 0
5         start = 0 # starting index of the current substring
6         for i in range(len(s)):
7             if s[i] not in s[start:i]:
8                 count = i - start + 1
9                 dic[count] = s[start:i + 1]
10            else:
11                start = s.find(s[i], start) + 1
12
13        if dic:
14            return max(dic.keys())
15        else:
16            return 0
```

Ln 14, Col 32 | Saved to local Run Submit

☒ Testcase | Test Result

Accepted

 Runtime: 29 ms

• Case 1

 • Case 2 • Case 3