

<https://leetcode.com/problems/two-sum>

← → ↺ leetcode.com/problems/two-sum/submissions/1335015154/

Problem List < > ↺ Run Submit ⌵

Description Accepted × Editorial Solutions Submissions

← All Submissions

Accepted

psharmapaul submitted at Jul 27, 2024 16:38

Editorial Solution

Runtime

88 ms | Beats 46.22%

Analyze Complexity

Memory

49.32 MB | Beats 84.72%

8% 6% 4% 2%

Code

```
1 /**
2  * @param {number[]} nums
3  * @param {number} target
4  * @return {number[]}
5  */
6 var twoSum = function(nums, target) {
7     for(let i=0;i<nums.length;i++){
8         for(let j=i+1;j<nums.length;j++){
9             if(nums[i]+nums[j]==target){
10                 return [i,j];
11             }
12         }
13     }
14 }
15
```

Saved Ln 1, Col 1

Testcase Test Result

Accepted Runtime: 52 ms

Case 1 Case 2 Case 3

<https://leetcode.com/problems/3sum>

← → ↺ leetcode.com/problems/3sum/submissions/1335041892/

Problem List < > ↺ Run Submit ⌵

Description Accepted × Editorial Solutions Submissions

← All Submissions

Accepted

psharmapaul submitted at Jul 27, 2024 17:13

Editorial Solution

Runtime

159 ms | Beats 56.89%

Analyze Complexity

Memory

65.22 MB | Beats 54.07%

40% 20% 0%

121ms 272ms 423ms 574ms 725ms 876ms 1027ms 1178ms

Code JavaScript

```
1 /**
2  * @param {number[]} nums
3  * @return {number[][]}
4  */
5 var threeSum = function (nums) {
6     nums.sort((a, b) => a - b);
7     const result = [];
8     for (let i = 0; i < nums.length - 2; i++) {
9         if (i > 0 && nums[i] === nums[i - 1]) continue;
10        let left = i + 1;
11        let right = nums.length - 1;
12        while (left < right) {
13            const sum = nums[i] + nums[left] + nums[right];
14            if (sum === 0) {
15                result.push([nums[i], nums[left], nums[right]]);
16                while (nums[left] === nums[left + 1]) left++;
17                while (nums[right] === nums[right - 1]) right--;
18                left++;
19                right--;
20            } else if (sum < 0) {
21                left++;
22            } else {
23                right--;
24            }
25        }
26    }
27    return result;
28 }
29
```

Saved Ln 19, Col 25

Testcase Test Result

<https://leetcode.com/problems/palindrome-number/>

leetcode.com/problems/palindrome-number/

Problem List < > < Run Submit < > Premium

Description Accepted x Editorial Solutions Submissi <

All Submissions

Accepted

psharmapaul submitted at Jul 27, 2024 17:17

Editorial Solution

Runtime

103 ms Beats 98.98%

Analyze Complexity

Memory

56.46 MB Beats 89.37%

6% 4% 2%

Code

```
1 /**
2  * @param {number} x
3  * @return {boolean}
4  */
5 var isPalindrome = function(x) {
6     if (x < 0) return false;
7     let original = x;
8     let reversed = 0;
9     while (x > 0) {
10         reversed = reversed * 10 + (x % 10);
11         x = Math.floor(x / 10);
12     }
13     return original === reversed;
14 }
15
```

Saved Ln 15, Col 3

Testcase Test Result

<https://leetcode.com/problems/maximum-subarray/>

leetcode.com/problems/maximum-subarray/submissions/1335046906/

Problem List < > < Run Submit < > Premium

Description Accepted x Editorial Solutions Submissions

All Submissions

Accepted

psharmapaul submitted at Jul 27, 2024 17:19

Editorial Solution

Runtime

62 ms Beats 92.60%

Analyze Complexity

Memory

58.02 MB Beats 28.69%

6% 4% 2%

Code

```
1 /**
2  * @param {number[]} nums
3  * @return {number}
4  */
5 var maxSubArray = function(nums) {
6     let maxSoFar = nums[0];
7     let maxEndingHere = nums[0];
8
9     for (let i = 1; i < nums.length; i++) {
10         maxEndingHere = Math.max(nums[i], maxEndingHere + nums[i]);
11         maxSoFar = Math.max(maxSoFar, maxEndingHere);
12     }
13
14     return maxSoFar;
15 }
16
```

Saved Ln 14, Col 21

Testcase Test Result