Problem List

Run Submit

Description | Editorial | Solutions | Submissions

53. Maximum Subarray

Attempted

Medium Topics Companies

Given an integer array `nums`, find the subarray with the largest sum, and return *its sum*.

Example 1:

```
Input: nums = [-2,1,-3,4,-1,2,1,-5,4]
Output: 6
Explanation: The subarray [4,-1,2,1] has the largest sum 6.
```

Example 2:

```
Input: nums = [1]
Output: 1
Explanation: The subarray [1] has the largest sum 1.
```

Example 3:

```
Input: nums = [5,4,-1,7,8]
Output: 23
Explanation: The subarray [5,4,-1,7,8] has the largest sum 23.
```

Constraints:

- `1 <= nums.length <= 10^5`
- `-10^4 <= nums[i] <= 10^4`

34K 236

Type here to search

Problem List | Run | Submit | 00:04:40 | Premium

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted Time taken: 4 m 40 s

Azan imtiaz submitted at Jul 14, 2024 13:59

Editorial | Solution

Runtime

1 ms | Beats 99.62%

Analyze Complexity

Memory

57.16 MB | Beats 23.72%

Code | Java

```java
class Solution {
    public int maxSubArray(int[] nums) {
        int sum=0;
        int max_sum=nums[0];
        for(int i=0;i<nums.length;i++){
            //step1
            sum=sum+nums[i];
            //step2
```

Code

Java | Auto

```java
class Solution {
    public int maxSubArray(int[] nums) {
        int sum=0;
        int max_sum=nums[0];
        for(int i=0;i<nums.length;i++){
            //step1
            sum=sum+nums[i];
            //step2
            if(max_sum < sum){
             max_sum=sum;
            }
            //step 3
            if(sum< 0) sum=0;
        }

        return max_sum;
    }
}
```

Saved | Ln 15, Col 9

Testcase | Test Result

nums =

[-2,1,-3,4,-1,2,1,-5,4]

Output

6

Type here to search | 34°C Haze | 2:01 PM 7/14/2024

<Dry Run 3>

arr = [5, -20, -20]

sum = 0 maxi = arr[0] = 5

i = 0

sum = 0 + (5) = 5; maxi = max(5,5) = 5; sum < 0 X

i = 1

sum = 5 + (-20) = -15; maxi = max(5, -15) = 5; sum < 0 V sum = 0

i = 2

sum = 0 + (-20) = -20; maxi = max(5, -20) = 5; sum < 0 V sum = 0

return 5 → maximumSum

<Day Run>

array = [5, 4, -1, 7, 8]

sum = 0; maxi = 5

i = 0

sum = 0 + 5 = 5; maxi = max(5, 5) = 5; sum < 0 X

i = 1

sum = 5 + 4 = 9; maxi = max(5, 9) = 9; sum < 0 X

i = 2

sum = 9 + (-1) = 8 maxi = max(9, 8) = 9; sum < 0 X

i = 3

sum = 8 + 7 = 15 maxi = max(9, 15) = 15 sum < 0 X

i = 4

sum = 15 + 8 = 23 max = max(15, 23) = 23; sum < 0 X

return 23 — maxSum

Problem List

Run Submit

Description Accepted Editorial Solutions Submissions

217. Contains Duplicate

Solved

Easy Topics Companies

Given an integer array `nums`, return `true` if any value appears **at least twice** in the array, and return `false` if every element is distinct.

Example 1:

```
Input: nums = [1,2,3,1]
Output: true
```

Example 2:

```
Input: nums = [1,2,3,4]
Output: false
```

Example 3:

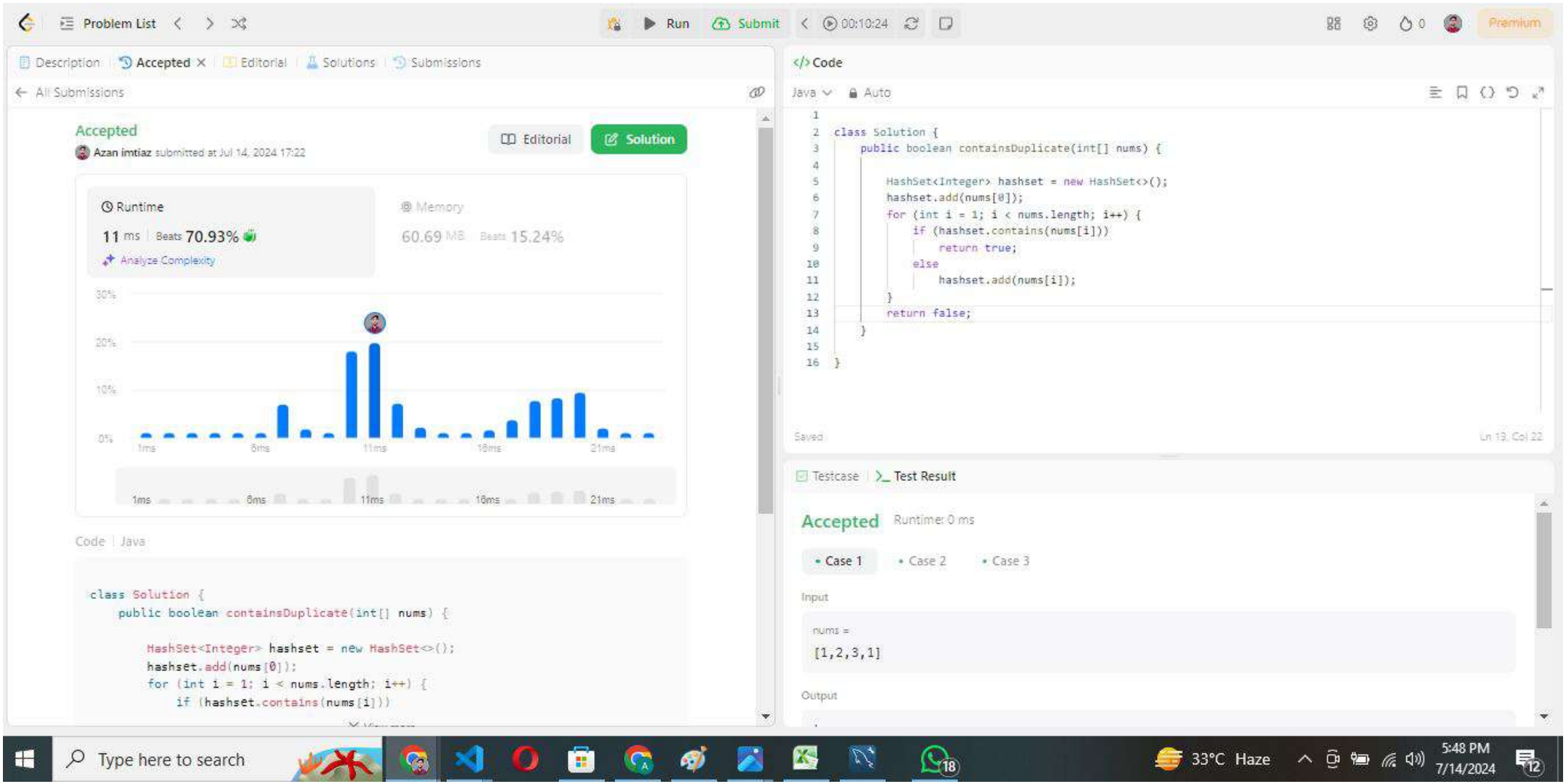
```
Input: nums = [1,1,1,3,3,4,3,2,4,2]
Output: true
```

Constraints:

- `1 <= nums.length <= 10^5`
- `-10^9 <= nums[i] <= 10^9`

12K 275

Type here to search


Problem List
Run
Submit
00:10:24
Premium
Description
Accepted
Editorial
Solutions
Submissions
All Submissions
Accepted
Azan imtiaz submitted at Jul 14, 2024 17:22
Editorial
Solution
Runtime
11 ms | Beats 70.93%
Analyze Complexity
Memory
60.69 MB | Beats 15.24%
30%
20%
10%
0%
1ms
6ms
11ms
16ms
21ms
Code | Java
class Solution {
public boolean containsDuplicate(int[] nums) {
HashSet<Integer> hashset = new HashSet<>();
hashset.add(nums[0]);
for (int i = 1; i < nums.length; i++) {
if (hashset.contains(nums[i]))
Code
Java
Auto
class Solution {
public boolean containsDuplicate(int[] nums) {
HashSet<Integer> hashset = new HashSet<>();
hashset.add(nums[0]);
for (int i = 1; i < nums.length; i++) {
if (hashset.contains(nums[i]))
return true;
else
hashset.add(nums[i]);
}
return false;
}
}
Saved
Ln 13, Col 22
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Case 3
Input
nums =
[1,2,3,1]
Output
Type here to search
33°C Haze
5:48 PM
7/14/2024

Dry Run >

nums = [1,2,3,1] Length = 4

Add first index element to hashSet variable v.

· v.add(nums[0]) = v.add(1) = [1] — it is not array

Loop

i = 1 0 < 4

if(v.contains(2)) X

v.add(2) so know v = [1,2] — not array

i = 2 0 < 4

if(v.contains(3)) X

v.add(3) so now v = [1,2,3] — not array

i = 3 3 < 4

if(v.contains(1)) ✓ return **true**

Time Complexity O(N) Space Complexity O(N)