Description | Accepted | Editorial | Solutions | Submissions

All Submissions

**Accepted** 112 / 112 testcases passed

SanjayChandagani submitted at Jan 08, 2025 10:38

Editorial | Solution

Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

43.83 MB | Beats 13.92%


Code | C++

```
//Sanjay Chandagani VU21CSEN00300372
class Solution {
public:
    int kadane(vector<int>& nums) {
        int currSum = nums[0];
        int maxSum = nums[0];
        for(int i = 1; i < nums.size(); i++) {
            currSum = max(nums[i], currSum + nums[i]);
```

View more

More challenges

- 2506. Count Pairs Of Similar Strings
- 2935. Maximum Strong Pair XOR II

Code

C++ | Auto

```
1  //Sanjay Chandagani VU21CSEN00300372
2  class Solution {
3  public:
4      int kadane(vector<int>& nums) {
5          int currSum = nums[0];
6          int maxSum = nums[0];
7          for(int i = 1; i < nums.size(); i++) {
8              currSum = max(nums[i], currSum + nums[i]);
9              maxSum = max(maxSum, currSum);
10         }
11         return maxSum;
12     }
13
14     int maxSubarraySumCircular(vector<int>& nums) {
15         // If all numbers are negative, return maximum element
16         bool allNegative = true;
17         for(int num : nums) {
18             if(num >= 0) {
19                 allNegative = false;
20                 break;
```

Ln 17, Col 29 | Saved

Run | Submit

Testcase | Test Result

**Accepted** Runtime: 0 ms

Case 1 | Case 2 | Case 3

Input

nums =

[1,-2,3,-2]

Output

3

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

**Accepted** 262 / 262 testcases passed

SanjayChandagani submitted at Jan 08, 2025 10:39

Editorial | Solution

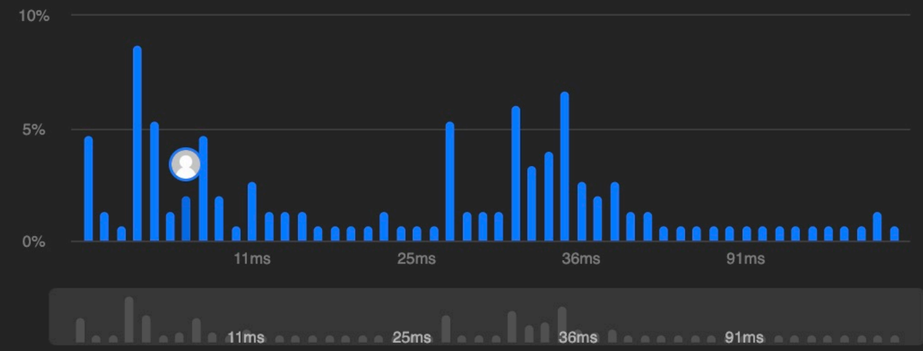
Runtime

6 ms | Beats 77.85%

Analyze Complexity

Memory

10.37 MB | Beats 91.28%




Code | C++

```
//Sanjay Chandagani VU21CSEN00300372
class Solution {
public:
    bool canReplace(string& target, int pos, string& stamp) {
        // Check if we can replace the window at pos with stamp
        for(int i = 0; i < stamp.length(); i++) {
            if(target[i + pos] != '?' && target[i + pos] != stamp[i]) {
                return false;
```

View more

More challenges

- 2616. Minimize the Maximum Difference of Pairs
- 2708. Maximum Strength of a Group

Code

C++ | Auto

```
1  //Sanjay Chandagani VU21CSEN00300372
2  class Solution {
3  public:
4      bool canReplace(string& target, int pos, string& stamp) {
5          // Check if we can replace the window at pos with stamp
6          for(int i = 0; i < stamp.length(); i++) {
7              if(target[i + pos] != '?' && target[i + pos] != stamp[i]) {
8                  return false;
9              }
10         }
11         return true;
12     }
13
14     int replace(string& target, int pos, int len, int count) {
15         // Replace the window with '?' and return number of new '?' added
16         int newCount = 0;
17         for(int i = 0; i < len; i++) {
18             if(target[pos + i] != '?') {
19                 target[pos + i] = '?';
20                 newCount++;
```

Ln 2, Col 1 | Saved

Run | Submit

Testcase | Test Result

**Accepted** Runtime: 0 ms

Case 1 | Case 2

Input

stamp =

"abc"

target =

"ababc"