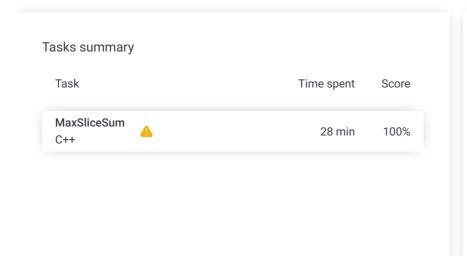
Codility_

CodeCheck Report: trainingFVDSKU-XA8

Test Name:

Check out Codility training tasks

Summary Timeline 🖨 Al Assistant Transcript





Tasks Details

1. MaxSliceSum

<u>=</u>asy

Find a maximum sum of a compact subsequence of array elements. Task Score

100%

Correctness

Performance

100%

100%

Task description

A non-empty array A consisting of N integers is given. A pair of integers (P, Q), such that $0 \le P \le Q < N$, is called a *slice* of array A. The *sum* of a slice (P, Q) is the total of A[P] + A[P+1] + ... + A[Q].

Write a function:

int solution(vector<int> &A);

that, given an array A consisting of N integers, returns the maximum sum of any slice of A.

For example, given array A such that:

$$A[0] = 3 \quad A[1] = 2 \quad A[2] = -6$$

$$A[3] = 4 \quad A[4] = 0$$

the function should return 5 because:

- (3, 4) is a slice of A that has sum 4,
- (2, 2) is a slice of A that has sum -6,
- (0, 1) is a slice of A that has sum 5,
- no other slice of A has sum greater than (0, 1).

Write an efficient algorithm for the following assumptions:

• N is an integer within the range [1..1,000,000];

Solution

Programming language used: C++

Total time used: 28 minutes

Effective time used: 28 minutes

Notes: not defined yet

Task timeline



show code in pop-up

final, score: 100

1 // you can use includes, for example:

- 2 // #include <algorithm>
- 3 #include <climits>

Code: 10:33:13 UTC, cpp,

- each element of array A is an integer within the range [-1,000,000..1,000,000];
- the result will be an integer within the range [-2,147,483,648..2,147,483,647].

Copyright 2009–2024 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

Test results - Codility

```
// you can write to stdout for debugging purpo
     // cout << "this is a debug message" << endl;</pre>
6
7
     int solution(vector<int>& A) {
         int maxslice = INT_MIN;
8
9
         int ending = INT_MIN;
10
         bool firstValue = true;
11
         for (int value : A) {
12
13
             if (firstValue) {
14
                 ending = value;
15
                 firstValue = false;
16
             } else {
17
                 ending = std::max(value, ending +
18
19
             maxslice = std::max(ending, maxslice);
20
21
22
         return maxslice;
23
```

Analysis summary

The solution obtained perfect score.

Analysis

Detected time complexity: O(N)

e tests
✓ OK
ess tests
✓ OK
nce tests
✓ OK
✓ OK
✓ OK
✓ OK
∠ OK