



Demo ticket

Session

ID: demoFZ4RSD-ZAJ
Time limit: 120 min.

Status: closed

Created on: 2014-03-16 03:11 UTC
Started on: 2014-03-16 03:16 UTC
Finished on: 2014-03-16 03:22 UTC

Tasks in test

Task score

Test score

?

100%

100 out of 100 points

MEDIUM

1. MaxSliceSum

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

score: 100 of 100



Task description

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

Write a function:

```
def solution(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  A[1] = 2  A[2] = -6
A[3] = 4  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).

Assume that:

- N is an integer within the range $[1..1,000,000]$;
- 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 $[-2,147,483,648..2,147,483,647]$.

Complexity:

- expected worst-case time complexity is $O(N)$;
- expected worst-case space complexity is $O(1)$, beyond input storage (not counting the storage required for input arguments).

Elements of input arrays can be modified.

Copyright 2009–2014 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

Solution

Programming language used: Python

Total time used: 7 minutes

Effective time used: 7 minutes

Notes: correct functionality and scalability

Task timeline



03:16:02

03:22:20

Code: 03:22:20 UTC, py, final, score: 100.00

```
1. def solution(A):
2.     # write your code in Python 2.6
3.     max_sofar = max_end = A[0]
4.     for i in xrange(1, len(A)):
5.         max_end = max(max_end + A[i], A[i])
6.         max_sofar = max(max_sofar, max_end)
7.     return max_sofar
```

Analysis

Detected time complexity:

$O(N)$

test	time	result
example	0.050 s.	OK
one_element	0.050 s.	OK
two elements	0.050 s.	OK

Codility

test_name	time	status
three_elements	0.050 s.	OK
simple	0.050 s.	OK
extreme_minimum	0.050 s.	OK
fifty_random	0.050 s.	OK
neg_const	0.050 s.	OK
pos_const	0.050 s.	OK
high_low_1Kgarbage	0.050 s.	OK
1Kgarbage_high_low	0.050 s.	OK
growing_saw	0.050 s.	OK
blocks	0.320 s.	OK
growing_negative	0.530 s.	OK

Training center