Codility_

CodeCheck Report: training8VDX9P-TH8

Test Name:

Check out Codility training tasks

Summary | Timeline

Tasks summary

Task	Time spent	Score
MaxSliceSum Python ⚠	8 min	100%

Total score

100%

Tasks Details

Easy

1. MaxSliceSum

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

Task Score	Correctness	Performance
100%	100%	100%

Task description

A non-empty array A consisting of N integers is given. A pair of integers (P, Q), such that 0 ≤ P ≤ 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:

```
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).

Write an **efficient** algorithm for the following assumptions:

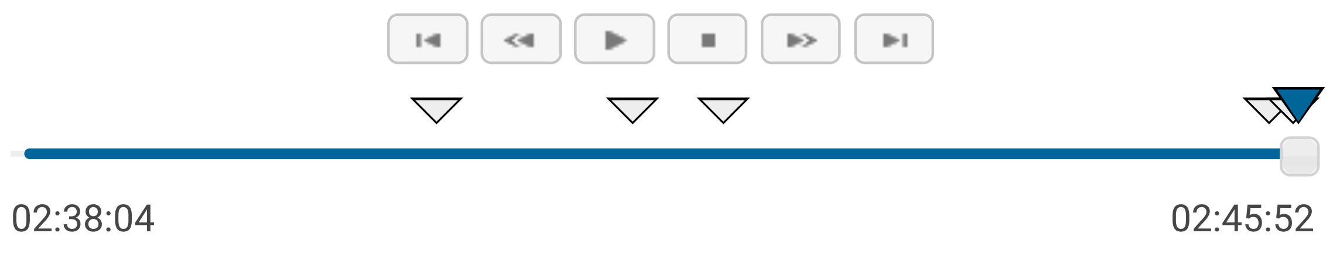
- N is an integer within the range [1..1,000,000];
- each element of array A is an integer within the range [−1,000,000..1,000,000];
- the result will be an integer within the range [−2,147,483,648..2,147,483,647].

Solution

Programming language used:	Python	
Total time used:	8 minutes	?
Effective time used:	8 minutes	?
Notes:	*not defined yet*	

Task timeline

?



Code: 02:45:51 UTC, py, final, score: 100

show code in pop-up

```python
# you can write to stdout for debugging purposes, e.g.
# print("this is a debug message")

def solution(A):
    # write your code in Python 3.6
    if not A:
        return None
    N = len(A)
    if N == 1:
        return A[0]
    if N == 2:
        return max(max(A[0],A[1]), A[0]+A[1])

    max_ = max_slice = min(A)*N
    sum_ = A[0]
    i = 0
    j = 0
    while True:
        if j < N-1 and i == j and A[j] <= 0:
            j += 1
            i = j
            max_ = sum_ = A[i]
        elif j < N and i == j and A[j] > 0:
            j += 1
            max_ = sum_ = A[i]
        elif j < N and sum_+A[j] >= 0:
            j += 1
            sum_ += A[j-1]
            max_ = max(max_, sum_)
        elif j < N and sum_+A[j] < 0:
            i = j+1
            j = i
        elif j == N:
            break
        max_slice = max(max_slice, max_)

    max_slice = max(max_slice, max_)
    return max_slice
```

Analysis summary

The solution obtained perfect score.

Analysis

Detected time complexity: **O(N)**

expand all	Example tests	
▶	example	✔ OK
expand all	**Correctness tests**	
▶	one_element	✔ OK
▶	two_elements	✔ OK
▶	three_elements	✔ OK
▶	simple	✔ OK
▶	extreme_minimum	✔ OK
▶	fifty_random	✔ OK
▶	neg_const	✔ OK
▶	pos_const	✔ OK
expand all	**Performance tests**	
▶	high_low_1Kgarbage	✔ OK
▶	1Kgarbage_high_low	✔ OK
▶	growing_saw	✔ OK
▶	blocks	✔ OK
▶	growing_negative	✔ OK