

## Demo ticket

### Session

ID: demo7U3Q7E-XQM  
Time limit: 120 min.

### Status: closed

Created on: 2014-03-16 02:56 UTC  
Started on: 2014-03-16 02:57 UTC  
Finished on: 2014-03-16 03:00 UTC

### Tasks in test

### Task score

Test score

?

100%

100 out of 100 points

MEDIUM

#### 1. MaxDoubleSliceSum

Find the maximal sum of any double slice.

score: 100 of 100



#### Task description

A non-empty zero-indexed array  $A$  consisting of  $N$  integers is given. A triplet  $(X, Y, Z)$ , such that  $0 \leq X < Y < Z < N$ , is called a *double slice*. The *sum* of double slice  $(X, Y, Z)$  is the total of  $A[X + 1] + A[X + 2] + \dots + A[Y - 1] + A[Y + 1] + A[Y + 2] + \dots + A[Z - 1]$ . For example, array  $A$  such that:

```
A[0] = 3
A[1] = 2
A[2] = 6
A[3] = -1
A[4] = 4
A[5] = 5
A[6] = -1
A[7] = 2
```

contains the following example double slices:

- double slice (0, 3, 6), sum is  $2 + 6 + 4 + 5 = 17$ ,
- double slice (0, 3, 7), sum is  $2 + 6 + 4 + 5 - 1 = 16$ ,
- double slice (3, 4, 5), sum is 0.

The goal is to find the maximal sum of any double slice. Write a function:

```
def solution(A)
```

that, given a non-empty zero-indexed array  $A$  consisting of  $N$  integers, returns the maximal sum of any double slice.

For example, given:

```
A[0] = 3
A[1] = 2
A[2] = 6
A[3] = -1
A[4] = 4
A[5] = 5
A[6] = -1
A[7] = 2
```

the function should return 17, because no double slice of array  $A$  has a sum of greater than 17.

#### Solution

Programming language used: Python

Total time used: 4 minutes

Effective time used: 4 minutes

Notes: correct functionality and scalability

#### Task timeline



Code: 03:00:47 UTC, py, final, score: 100.00

```
01. def solution(A):
02.     # write your code in Python 2.6
03.     size = len(A)
04.     prefix = [0]*size
05.     for i in xrange(1, size):
06.         prefix[i] = max(0, A[i], prefix[i-1]+A[i])
07.
08.     suffix = [0]*size
09.     for i in xrange(size-2, -1, -1):
10.         suffix[i] = max(0, A[i], suffix[i+1]+A[i])
11.
12.     max_slice = 0
13.     for i in xrange(1, size-1):
14.         max_slice = max(max_slice, prefix[i-1] +
15.                         suffix[i+1])
16.     return max_slice
```

#### Analysis

Assume that:

- N is an integer within the range [3..100,000];
- each element of array A is an integer within the range [-10,000..10,000].

Complexity:

- expected worst-case time complexity is  $O(N)$ ;
- expected worst-case space complexity is  $O(N)$ , 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.

Detected time complexity:

**$O(N)$**

test	time	result
example example test	0.050 s.	OK
simple1 first simple test	0.050 s.	OK
simple2 second simple test	0.050 s.	OK
simple3 third simple test	0.050 s.	OK
negative all negative numbers	0.050 s.	OK
positive all positive numbers	0.050 s.	OK
extreme_triplet three elements	0.050 s.	OK
small_random1 random, numbers from $-10^{**4}$ to $10^{**4}$ , length = 70	0.050 s.	OK
small_random2 random, numbers from -30 to 30, length = 300	0.050 s.	OK
medium_range -1000, ..., 1000	0.050 s.	OK
large_ones random numbers from -1 to 1, length = ~100,000	0.350 s.	OK
large_random random, length = ~100,000	0.370 s.	OK
extreme_maximal all maximal values, length = ~100,000	0.360 s.	OK
large_sequence many the same small sequences, length = ~100,000	0.350 s.	OK

Training center