

AVAILABLE  
LESSONS:Lesson 1  
IterationsLesson 2  
ArraysLesson 3  
Time ComplexityLesson 4  
Counting  
ElementsLesson 5  
Prefix SumsLesson 6  
SortingLesson 7  
Stacks and  
QueuesLesson 8  
LeaderLesson 9  
Maximum slice  
problemLesson 10  
Prime and  
composite  
numbers

Lesson 11

PAINLESS

## MaxNonoverlappingSegments

START

Find a maximal set of non-overlapping segments.

Programming language: C++

Located on a line are  $N$  segments, numbered from  $0$  to  $N - 1$ , whose positions are given in zero-indexed arrays  $A$  and  $B$ . For each  $I$  ( $0 \leq I < N$ ) the position of segment  $I$  is from  $A[I]$  to  $B[I]$  (inclusive). The segments are sorted by their ends, which means that  $B[K] \leq B[K + 1]$  for  $K$  such that  $0 \leq K < N - 1$ .

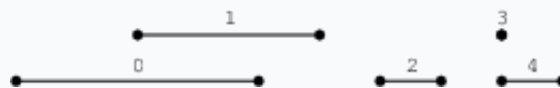
Two segments  $I$  and  $J$ , such that  $I \neq J$ , are *overlapping* if they share at least one common point. In other words,  $A[I] \leq A[J] \leq B[I]$  or  $A[J] \leq A[I] \leq B[J]$ .

We say that the set of segments is *non-overlapping* if it contains no two overlapping segments. The goal is to find the size of a non-overlapping set containing the maximal number of segments.

For example, consider arrays  $A, B$  such that:

$A[0] = 1$	$B[0] = 5$
$A[1] = 3$	$B[1] = 6$
$A[2] = 7$	$B[2] = 8$
$A[3] = 9$	$B[3] = 9$
$A[4] = 9$	$B[4] = 10$

The segments are shown in the figure below.



The size of a non-overlapping set containing a maximal number of segments is 3. For example, possible sets are  $\{0, 2, 3\}$ ,  $\{0, 2, 4\}$ ,  $\{1, 2, 3\}$  or  $\{1, 2, 4\}$ . There is no non-overlapping set with four segments.

Write a function:

```
int solution(vector<int> &A, vector<int> &B);
```

Sieve of  
Eratosthenes

*Lesson 12*  
Euclidean  
algorithm

*Lesson 13*  
Fibonacci  
numbers

*Lesson 14*  
Binary search  
algorithm

*Lesson 15*  
Caterpillar  
method

*Lesson 16*  
**Greedy  
algorithms**

*Lesson 17*  
Dynamic  
programming

*Lesson 90*  
Tasks from  
Indeed Prime  
2015 challenge

*Lesson 91*  
Tasks from  
Indeed Prime  
2016 challenge

*Lesson 92*  
Tasks from  
Indeed Prime  
2016 College  
Coders  
challenge

*Lesson 99*

that, given two zero-indexed arrays A and B consisting of N integers, returns the size of a non-overlapping set containing a maximal number of segments.

For example, given arrays A, B shown above, the function should return 3, as explained above.

Assume that:

- N is an integer within the range [0..30,000];
- each element of arrays A, B is an integer within the range [0..1,000,000,000];
- $A[I] \leq B[I]$ , for each I ( $0 \leq I < N$ );
- $B[K] \leq B[K + 1]$ , for each K ( $0 \leq K < N - 1$ ).

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 2009–2016 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

**For  
programmers**

Lessons Challenges  
Terms FAQ

**For companies**

Tour About  
Pricing us  
Blog Jobs  
Privacy Terms  
Cookies  
API

**Sign up for our  
newsletter:**

Information about  
upcoming challenges,  
solutions and lessons  
directly in your inbox.

**Social:**

[f](#) [t](#) [in](#)

**Contact us:**

For customer support  
queries:

UK +44 (0) 208 970  
78 68

US 1-415-466-8085  
support@codility.com

For sales queries:

UK +44 (0) 208 970  
78 67

US 1-415-466-8085  
sales@codility.com

---

© 2009–2016 Codility Ltd., registered in England and Wales (No. 7048726). VAT ID  
GB981191408. Registered office: 107 Cheapside, London EC2V 6DN