

AVAILABLE
LESSONS:*Lesson 1*
Iterations*Lesson 2*
Arrays*Lesson 3*
Time Complexity*Lesson 4*
Counting
Elements*Lesson 5*
Prefix Sums*Lesson 6*
Sorting*Lesson 7*
Stacks and
Queues*Lesson 8*
Leader*Lesson 9*
Maximum slice
problem*Lesson 10*
Prime and
composite
numbers*Lesson 11*

AMBITIOUS

NailingPlanks

START

Count the minimum number of nails that allow a series of planks to be nailed.

Programming language: C++ ▼

You are given two non-empty zero-indexed arrays A and B consisting of N integers. These arrays represent N planks. More precisely, A[K] is the start and B[K] the end of the K-th plank.

Next, you are given a non-empty zero-indexed array C consisting of M integers. This array represents M nails. More precisely, C[I] is the position where you can hammer in the I-th nail.

We say that a plank (A[K], B[K]) is nailed if there exists a nail C[I] such that $A[K] \leq C[I] \leq B[K]$.

The goal is to find the minimum number of nails that must be used until all the planks are nailed. In other words, you should find a value J such that all planks will be nailed after using only the first J nails. More precisely, for every plank (A[K], B[K]) such that $0 \leq K < N$, there should exist a nail C[I] such that $I < J$ and $A[K] \leq C[I] \leq B[K]$.

For example, given arrays A, B such that:

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

four planks are represented: [1, 4], [4, 5], [5, 9] and [8, 10].

Given array C such that:

C[0] = 4
C[1] = 6
C[2] = 7
C[3] = 10
C[4] = 2

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

if we use the following nails:

- 0, then planks [1, 4] and [4, 5] will both be nailed.
- 0, 1, then planks [1, 4], [4, 5] and [5, 9] will be nailed.
- 0, 1, 2, then planks [1, 4], [4, 5] and [5, 9] will be nailed.
- 0, 1, 2, 3, then all the planks will be nailed.

Thus, four is the minimum number of nails that, used sequentially, allow all the planks to be nailed.

Write a function:

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

that, given two non-empty zero-indexed arrays A and B consisting of N integers and a non-empty zero-indexed array C consisting of M integers, returns the minimum number of nails that, used sequentially, allow all the planks to be nailed.

If it is not possible to nail all the planks, the function should return -1.

For example, given arrays A, B, C such that:

```
A[0] = 1    B[0] = 4
A[1] = 4    B[1] = 5
A[2] = 5    B[2] = 9
A[3] = 8    B[3] = 10
```

```
C[0] = 4
C[1] = 6
C[2] = 7
C[3] = 10
C[4] = 2
```

the function should return 4, as explained above.

Assume that:

- N and M are integers within the range [1..30,000];
- each element of arrays A, B, C is an integer within the range [1..2*M];
- $A[K] \leq B[K]$.

Complexity:

- expected worst-case time complexity is $O((N+M)*\log(M))$;
- expected worst-case space complexity is $O(M)$, beyond input storage (not counting the storage

[Future training](#)

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

[About us](#)
[Tour](#) [Jobs](#)
[Pricing](#) [Terms](#)
[Blog](#) [Privacy](#) [Cookies](#)
[API](#)

Sign up for our newsletter:

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

[Sign up](#)

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