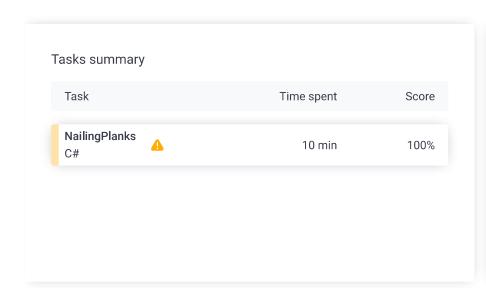
Codility_

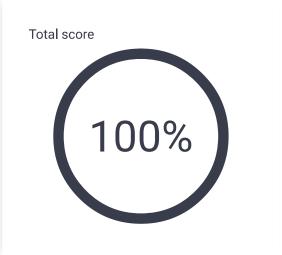
CodeCheck Report: trainingBUA7M6-TZ4

Test Name:

Summary

Timeline





Check out Codility training tasks

Tasks Details

Medium

1. NailingPlanks

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



Task description

You are given two non-empty 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 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 $A[K] \le C[I] \le B[K]$.

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 $0 \le K < N$, there should exist a nail C[I] such that 1 < J and A[K] $\le C[I] \le B[K]$.

For example, given arrays A, B such that:

Solution

Programming language used:	C#	
Total time used:	10 minutes	•
Effective time used:	10 minutes	•
Notes:	not defined yet	
Task timeline	H >> 1	3
07:52:07		08:02:00

```
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
```

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:

```
class Solution { public int solution(int[] A,
int[] B, int[] C); }
```

that, given two non-empty arrays A and B consisting of N integers and a non-empty 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.

Write an efficient algorithm for the following assumptions:

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

Copyright 2009–2023 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

```
score: 100
1
     using System;
2
3
      * 14.2 - Nailing Planks
4
5
      * Paulo Santos
6
      * 11.Jan.2023
7
8
     class Solution {
9
         public int solution(int[] A, int[] B, int[] C)
10
11
             var start = 0;
12
             var end = C.Length;
             var midNails = -1;
13
14
             while (start <= end) {
15
                  var mid = (start + end) / 2;
16
                  if (CheckNail(A, B, C, mid)) {
17
                      end = mid - 1;
                      midNails = mid;
18
19
                  }
20
                  else
                      start = mid + 1;
21
22
             }
23
             return midNails;
24
         }
25
         private bool CheckNail(int[] A, int[] B, int[]
26
27
28
             var M = C.Length;
29
             var prefixSum = new int[2 * M + 1];
30
             for (var i = 0; i < mid; i++)
                  prefixSum[C[i]] += 1;
31
32
             for (var i = 1; i < prefixSum.Length; i ++)</pre>
33
                  prefixSum[i] += prefixSum[i - 1];
34
             for (var i = 0; i < A.Length; i++)
                  if (prefixSum[B[i]] == prefixSum[A[i] -
35
36
                      return false;
37
38
             return true;
39
40
         }
41
     }
```

show code in pop-up

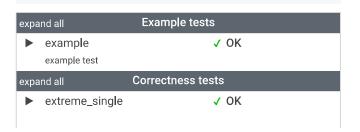
Code: 08:02:00 UTC, cs, final,

Analysis summary

The solution obtained perfect score.

Analysis

```
Detected time complexity: O((N + M) * log(M))
```



single	e nail and single plank	
>	extreme_point nail is a point [1, 1]	√ OK
>	few_nails_in_the_same_place few nails are in the same place	√ OK
>	random_small random sequence, length = ~100	√ OK
ехра	nd all Performance t	ests
	random_medium	✓ OK
	random sequence, length = ~10,000	
>	_	✓ OK
>	random sequence, length = ~10,000 random_large	✓ OK