

Tasks Details

[Check out Codility training tasks](#)

Medium

1. CountNonDivisible

Calculate the number of elements of an array that are not divisors of each element.

Task Score

100%

Correctness

100%

Performance

100%

Task description

You are given an array A consisting of N integers.

For each number $A[i]$ such that $0 \leq i < N$, we want to count the number of elements of the array that are not the divisors of $A[i]$. We say that these elements are non-divisors.

For example, consider integer $N = 5$ and array A such that:

```
A[0] = 3
A[1] = 1
A[2] = 2
A[3] = 3
A[4] = 6
```

For the following elements:

- $A[0] = 3$, the non-divisors are: 2, 6,
- $A[1] = 1$, the non-divisors are: 3, 2, 3, 6,
- $A[2] = 2$, the non-divisors are: 3, 3, 6,
- $A[3] = 3$, the non-divisors are: 2, 6,
- $A[4] = 6$, there aren't any non-divisors.

Write a function:

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

that, given an array A consisting of N integers, returns a sequence of integers representing the amount of non-divisors.

Result array should be returned as an array of integers.

For example, given:

```
A[0] = 3
A[1] = 1
A[2] = 2
A[3] = 3
A[4] = 6
```

the function should return [2, 4, 3, 2, 0], as explained above.

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

- N is an integer within the range [1..50,000];
- each element of array A is an integer within the range [1..2 * N].

Solution

Programming language used: C#

Total time used:

4 minutes

?

Effective time used:

4 minutes

?

Notes:

not defined yet

Task timeline

?



13:42:41

13:45:53

Code: 13:45:52 UTC, cs, final,
score: 100

[show code in pop-up](#)

```
1  using System;
2  using System.Linq;
3  using System.Collections.Generic;
4
5  /**
6   * 11.1 - Count Non Divisible
7   */
8  class Solution {
9      public int[] solution(int[] A) {
10
11
12          var ans = new int[A.Length];
13
14          /*
15           * Count the number of occurrences
16           */
17          var max = A.Max();
18          var cnt = new int[max + 1];
19          for(var i = 0; i < A.Length; i++)
20              cnt[A[i]] += 1;
21
22          /*
23           * Count the number of divisors
24           */
25          for (int i = 0; i < A.Length; i++)
26              {
```

```
27      /*
28       * Calculate how many of its divisors
29       * are in the array
30       */
31      int divisors = 0;
32
33      for (int j = 1; j * j <= A[i]; j++)
34          if (A[i] % j == 0)
35          {
36              divisors += cnt[j];
37              if (A[i] / j != j)
38                  divisors += cnt[A[i] / j];
39          }
40
41      /*
42       * Subtract the number of divisors
43       * from the number of elements in the array
44       */
45      ans[i] = A.Length - divisors;
46  }
47
48  return ans;
49  }
50  }
```

Analysis summary

The solution obtained perfect score.

Analysis

Detected time complexity:

$O(N * \log(N))$

expand all		Example tests
▶	example	✓ OK
		example test
expand all		Correctness tests
▶	extreme_simple	✓ OK
		extreme simple
▶	double	✓ OK
		two elements
▶	simple	✓ OK
		simple tests
▶	primes	✓ OK
		prime numbers
▶	small_random	✓ OK
		small, random numbers, length = 100
expand all		Performance tests
▶	medium_random	✓ OK
		medium, random numbers length = 5,000
▶	large_range	✓ OK
		1, 2, ..., N, length = ~20,000

▶ **large_random** ✓ **OK**

large, random numbers, length =
~30,000

▶ **large_extreme** ✓ **OK**

large, all the same values, length =
50,000