Codility_

CodeCheck Report: training7WUKUZ-Z36

Test Name:

Check out Codility training tasks

Summary   Timeline   AI Assistant Transcript

Tasks summary

Task	Effective time spent	Score
CountNonDivisible ⚠ C++20	9 min	100%

Total score

100%

Tasks Details

		Task Score	Correctness	Performance
Medium	1. **CountNonDivisible** Calculate the number of elements of an array that are not divisors of each element.	100%	100%	100%

Task description

You are given an array A consisting of N integers.

For each number A[i] such that 0 ≤ 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
```

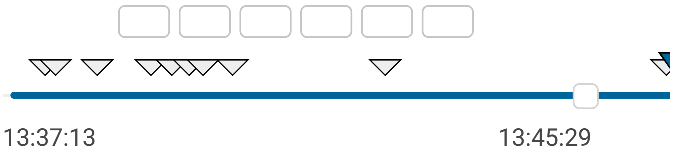
For the following elements:

- A[0] = 3, the non-divisors are: 2, 6,
- A[1] = 1, the non-divisors are: 3, 2, 3, 6,

Solution

Programming language used:	C++20
Time spent on task:	9 minutes
Notes:	*not defined yet*

Task timeline



- 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:

```
vector<int> solution(vector<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].

Copyright 2009–2024 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

Code: 13:45:29 UTC, cpp_20, final, score: **100**

show code in pop-up

```
// you can use includes, for example:
// #include <algorithm>
#include <unordered_map>
#include <vector>
// you can write to stdout for debugging pu
// cout << "this is a debug message" << end

using namespace std;

vector<int> solution(vector<int> &A) {
  int N = A.size();
  unordered_map<int, int> count;

// 1
  for (int i : A) {
    count[i]++;
  }

// 2
  vector<int> divisors(2 * N + 1, 0);
  for (auto &[i, count] : count) {
    for (int j = i; j <= 2 * N; j += i) {
      divisors[j] += count;
    }
  }

// 3
  vector<int> answer;
  for (int i : A) {
    answer.push_back(N - divisors[i]);
  }

  return answer;
}
```

Analysis summary

The solution obtained perfect score.

Analysis

Detected time complexity: **O(N * log(N))**

expand all	Example tests	
▶	example example test	✔ OK
expand all	Correctness tests	
▶	extreme_simple extreme simple	✔ OK
▶	double two elements	✔ OK
▶	simple simple tests	✔ OK
▶	primes prime numbers	✔ OK

▶ 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