

Find Median

2 seconds, 256 megabytes

We have work for you again, a very talented programmer.

You are assigned to find a median of an unsorted data with the length of N . However, the size of it is very big and your client is okay to have some error. As long as it is in the acceptable range

For the data set with N numbers where each number N_i follows $-2^{63} \leq N_i \leq 2^{63} - 1$, your accuracy (a.k.a score S) is $(1 - \frac{|P_x - 50|}{100})^2$ where P_x is percentile of your guessed value in the data set. Your score is 0 if the answer is not in the range of the data.

You are graded correct if summation of S across all T questions are greater or equal to C

Function

You need to implement the following function

```
int64_t find_median(std::vector<int64_t> nums);
```

Your function will be called T times with different data and your function must not read from stdin or print to stdout.

Warning Your code should not exceed time limit, despite your score may surpass C , it will still be graded as "Time Limit Exceeded".

Scoring

There are 10 test cases 10 scores each

General Constraints: $T \leq 100$, $N \leq 1000000$, $C \geq 0.8 * T$

10 Score: $N \leq 10000$, $C \geq 0.7 * T$

10 Score: $C \geq 0.5 * T$

10 Score: $T \leq 10$

10 Score: Range of all data sets does not exceed 100000

60 Score: No additional constraint

Your scores are checked during the runtime but it is guarantee that the time used to check will not exceed $O(\log N)$ basically, it will not impact your program.

Sample Grader Input

First Line contains T, C representing the number of questions and passing score

For each T questions each question has 3 lines

First Line of question contains N representing the length of the data

Second and Third Line of question contains the data set itself, one is randomized and the other is sorted. **The two data sets must be identical**

Examples

`find_median([9, 4, 5, 2, 1, 6, 3, 8, 7])` should returns 5

Returning 5 gives full 1 Score

While returning 4 or 6 gives score of 0.81 because $(1 - \frac{|40-50|}{100})^2 = 0.81$

and returning 3 or 7 gives score of 0.64 , if the C value is more than this, this will be graded incorrect.

So, make sure to keep your algorithm as accurate as possible and keep in mind the input size and time limit.