

Amazon Interview Experience

- Last Updated : 06 Sep, 2019

Round 1: Online test

You have M identical ropes and N identical weights. You want to do an experiment to determine the strength S of the identical ropes, by testing how many weights can be held by one single rope. The strength S of the ropes is defined as:

- 1) If a rope breaks with the first weight, $S=0$;
- 2) If a rope can hold n weights and breaks at $n+1$ weights, $S=n$;
- 3) If a rope can hold all N weights, $S=N$.

In one test (test means to check if one rope can hold an amount of weights), if the rope breaks, you have to take another rope to continue the experiment; if the rope does not break, it can be used for the next test with no problem. If you used up all the ropes but still cannot determine the strength S of the ropes, the experiment fails.

Please write a C/C++ program that, given M and N (M and N are both integers, $M \geq 1$, $N \geq 1$), calculates the minimum number of tests T needed to guarantee you can determine S . Try to optimize the time complexity of your program, and explain:

- 1) What kind of tricks you have used to optimize the time complexity?
- 2) What is the time complexity without these tricks and what is the time complexity with these tricks?

Hint 1: If you have limited number of ropes, you don't dare to take the risk. For example, if $M=1$, your only choice is to increase the weights one by one, from 1 to N , to make sure you can determine S in the worst case. In this case, $T=N$.

Hint 2: If you have lots of ropes, binary search is obviously helpful to calculate the minimum T efficiently.

Example 1: $M=1$, $N=20$

Command: `calc_n_tests 1 20`

Output: 20

Example 2: $M=2$, $N=5$

Command: `calc_n_tests 2 5`

Output: 3

My Personal Notes

Add your personal notes here

Save