

University of São Paulo
Institute of Mathematics and Statistics

VIII USP Freshman Try-outs
Editorial

Problem	Author	Prepared by
A - Agorabusiness	Marcelo Lage	Marcelo Lage
B - Balloon Quantum Popping	Matheus Paulo	Willian Mori
C - Coffee Break	Gabriel Morete	Gabriel Morete
D - Divine Music	Enrique Junchaya	Enrique Junchaya
E - Enigma of the Sphinx	Pedro Sousa	Pedro Sousa
F - Fierce Election	Marcelo Lage	Nathan Luiz
G - Gusteseu and Maynotauro	Matheus Paulo	Matheus Paulo
H - Hero Morethor	Thiago Oliveira	Otávio de Oliveira
I - Initial Ideas	Thiago Oliveira	Thiago Oliveira
J - Joyful Feast of the Gods	Enrique Junchaya	Pedro Simões

Table 1: Contest problems.

A. Agorabusiness.

Tags : Math.

Let's solve the problem for only one market. Notice the goal of Renang and Wan is to buy an even number of fruits.

If the same selling point v sells both even-sized and odd-sized packets the answer will be “SIM” (they can buy anything in the other selling points and, in selling point v , choose a package of the parity that will leave the total sum even).

Otherwise, all selling points sell either all even-sized or odd-sized packages. So it remains to verify whether the number of selling points selling only odd-sized packets is even (output “YES”) or odd (output “NAO”).

This solution runs in $O(p)$ if p is the total number of packets in all markets.

B. Balloon Quantum Popping.

Tags : Greddy, data structures.

We observe that in order to reach the minimum time required for the number of inflated balloons to be less than S , an intuitive strategy that ensures achieving this is always to look at the color of the balloon appearing the most. This type of strategy is called a greedy strategy (because we only look at the current state — in this case, the number of balloons of each color — to decide what to

do). For this purpose, there are data structures like *priority_queue* and *multiset*, which are capable of extracting the largest number from a collection of numbers (which can be modified).

C. Coffee Break.

Tags : Implementation, brute force.

Brute force: we can test each possible rectangular subsection of the table directly. There are $O((nm)^2)$ such subsections, and checking if each one of them is composed only of 1s can be done in $O(nm)$ by verifying all cells one by one (it's possible to do this more efficiently in general, but it won't be necessary in this problem).

The total complexity is $O((nm)^3)$, which is sufficient because $nm \leq 400$.

D. Divine Music.

Tags : Implementation, brute force.

Notice that, if we fix the first two numbers of the sequence, the rest would be determined. Hence, we have just 9 possibilities and, in fact, two of them are not valid (it's not possible to start with "0 0" or with "2 2"). Therefore, it suffices to test all the 7 valid possibilities and test if it matches with the given sequence.

The easiest way to do that is using two **for** loops: the first one to fix the first number of the sequence and the second one to fix the second number. Then, we can build the unique sequence determined by the fixed numbers. If it's possible to get the input sequence after blurring some of the digits of the built sequence, then we should add +1 to our answer.

E. Enigma of the Sphinx.

Tags : Math, ad-hoc.

We can rewrite the desired final situation as $\frac{f}{v+f} = \frac{3}{4}$ or, alternatively, $3v = f$.

If initially $3v - f < 0$, the statement will be true, and it will remain true until $3v - f \geq 0$, which will happen $\left\lceil \frac{f-3v}{3} \right\rceil$ times, as each time the value of $3v - f$ increases by 3.

Hence, it is sufficient to solve for when the initial values of v and f are such that $3v - f \geq 0$. The answer will then be exactly the value of $3v - f$, because the claim "more than 75% of statistics are false" will be false, and $3v - f$ decreases by 1 at a time until it reaches 0.

F. Fierce Election.

Tags : Binary search.

The idea is to perform a binary search on the answer.

In other words, consider the property

$$P(N) : \text{Kobus can win by buying } N \text{ votes}$$

It's clear that if $P(N)$ holds for a certain N , then $P(M)$ holds for all $M > N$. Therefore, if we want to find the smallest N for which $P(N)$ holds, we can perform a binary search on N , in no more than $O(\log(10^9 m))$ steps.

So, let's ask ourselves something more concrete: "given a quantity x of votes, can Kobus win by buying at most x votes?"

And this can be easily verified in $O(m)$ time, as it's sufficient to check:

$$\sum_{i=1}^n \max(0, a_i - (a_0 + x - 1)) \leq x$$

The complete solution runs in $O(m \log(10^9 m))$ time.

G. Gusteseu and Maynotauro.

Tags: Math.

An important fact is that the number of ways to traverse an $m \times n$ grid from its top-left corner to the bottom-right corner, using the least possible number of steps (i.e., only using $\downarrow$ and $\rightarrow$ steps), is given by $\binom{n+m-2}{n-1}$.

This is true because each of these paths can be seen as an anagram of $\underbrace{\downarrow \cdots \downarrow}_{m-1} \underbrace{\rightarrow \cdots \rightarrow}_{n-1}$.

Hence, the number of paths on a grid from $(1,1)$ to (m,n) without passing through (x,y) is equal to $\binom{n+m-2}{n-1}$ minus the number of paths that pass through (x,y) . However, to pass through (x,y) , the set of the first $x+y-2$ movements is determined (as well as the last $n+m-x-y$ movements), and thus this number will be $\binom{x+y-2}{x-1} \cdot \binom{n+m-x-y}{n-x}$.

For the binomial coefficients, it's possible to precompute them in a matrix using Stifel's relation and then query them in $O(1)$.

H. Hero Morethor.

Tags : Implementation.

It is important to note that Morethor faces the monsters (possibly none) in the order they are encountered, that is, in the order given by the input. Therefore, sorting the values of F_i is an incorrect approach. We can maintain a variable X that stores our hero's current strength, initially set to the ability H provided. We can then iterate through each of the monster's strength values in the order they were given, checking in the i -th iteration if Morethor's current strength value is greater than or equal to F_i , i.e., $X \geq F_i$. If it is, we update X by setting $X = X + F_i$ and keep going. Otherwise, the answer is "NAO" ("NO" in portuguese). If we reach the end of the sequence and the condition is met, the answer is "SIM" ("YES" in portuguese).

I. Initial Ideas.

Tags: Ad-hoc, implementation.

Note that, given a word from the tablet, there is at most one word from U that can correspond to it. For example, in case 1, PWP could only correspond to OVO (not VOO or TLE).

Therefore, the idea is to maintain the final permutation considering that each new word read "forces" some values in this permutation. In the second example test case, for example, we start with:

$$attempt_permutation = \text{"\$"}$$

where \$ represents an unfilled position.

BD can only correspond to AC, so we update *attempt_permutation*:

attempt_permutation = “B\$D\$”

CMOR can only correspond to AMOR, but the position is already occupied with a different letter than the one we want to place there now. We conclude that the permutation does not exist.

In the fourth example test case, after reading all the strings, we would have:

attempt_permutation = “A\$”

Now it’s just a matter of checking if it’s possible to fill the unfilled \$ positions in a way that the total string becomes a permutation.

There are various ways to do this, for example using a set: maintain a set with all the letters of the alphabet, initially iterate to remove from it all the letters that already have a forced position in the permutation (and during this iteration, also check if any letter appears twice, which means that the desired permutation does not exist), and then iterate again to fill the positions marked \$ with the remaining elements from the set.

J. Joyful Feast of the Gods.

Tags: Math.

According to the statement, each meal at the university restaurant costs two drachmas, so, if d drachmas had been used, one can infer that Nathena has had $\frac{d}{2}$ meals. It is also worth noting that, as x and y are always of the same parity, $d = x - y$ is always divisible by 2. So, denoting by x the value of the refill and by y the current balance, the number of times Nathena has feasted at the university restaurant is given by $\frac{d}{2} = \frac{x-y}{2}$.