Interactive Problems

Codeforces

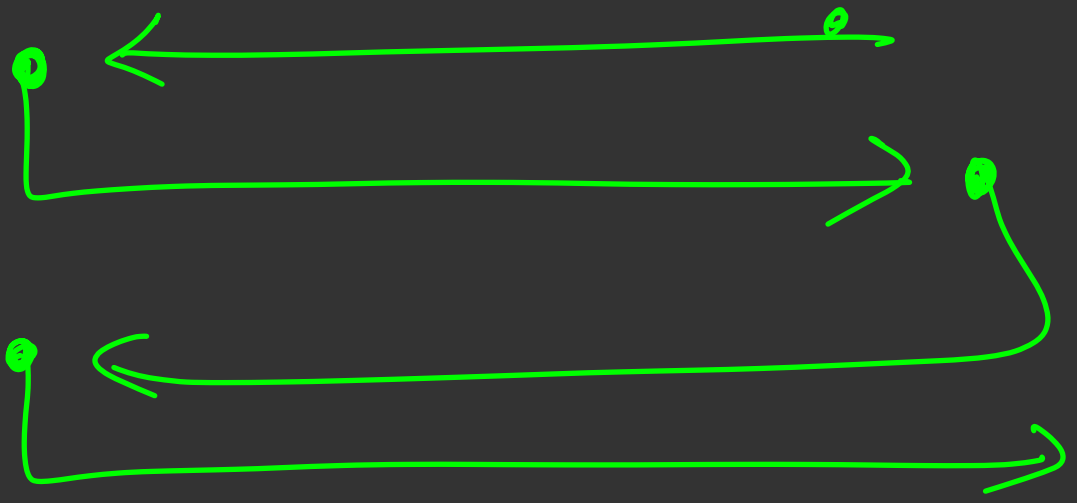
Sum of all elements

n elements

entire input ⟶ processing → output

Your program

Codeforces



- Makes you do stuff ONLINE.
- Read the problem clearly, understand the constraints, number of queries allowed and then come up with an algorithm that gets you the answer quickly without exceeding those constraints.
- Flush the output after every print statement. Link

Problem:

There is a hidden number. You can guess a number and the computer will tell you if the number is equal to the hidden number, greater or smaller. Find the number quickly. When found, print ! X as your answer.

Constraints: 1 <= N <= 100, You're allowed to ask at most 10 queries

Example Query: If the hidden number is 10

? 2

< The computer replies with '<' because clearly 2 < 10.

? 10

= The computer replies with '=' because 10 = 10

$\log_2(100) \leq 7$

Hidden Number = 10

Your query	? 2	2 is less than 10
Computer's Answer	<	
Your query	? 8	8 is still less than 10
Computer's Answer	<	
Your query	? 11	11 is greater than 10
Computer's Answer	>	
Your query	? 10	Found it
Computer's Answer	=	
Printing the Final Answer	! 10	

```
while ( true ) {
    int x = random(1,100)
    cout << "? " << x <<endl;
    char codeforces
    cin >> codeforces
    if ( codeforces == "=" ) break;
}
```

Normal

1. Time
2. Space

Interactive

1. Time
2. Space
3. Limit on queries | interactions

Input / output

0.1 seconds

```
for( int i=0 ; i<1000 ; i++) {

cout << 1; --> 0.5 seconds

}
```

(0.6) x 1000

1 1 1 1 1 1 ——— 1000 times

$(0.1) \times 1000 + \frac{1}{10} \times 1000 \times 0.5$

Buffer

① `cout << 1;` heavy task

② `c = a + b;`

```
for (int i=0; i<n; i++)
    cout << i << "\n";
```

① — character

does not flush

```
for (int i=0; i<n; i++
    cout << i << endl;
```

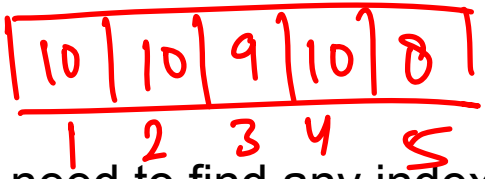
②

flushes

```
for (int i=0 ; i<10 ; i++)
    cout << 1 << endl;  "\n";
```

Problem:

limit on queries = 20



There is a hidden array of N integers, you need to find any index of the majority element or report that no such element exists. You can ask for any index and the computer will tell you if that index contains the majority element or not.

Constraints: $1 <= N <= 10^{18}$

length of hidden Array $> n/2$ times

Example Query: Hidden array [10, 11, 10]

$1 \le arr[i] \le 10^{18}$

? 2

index = 3 no

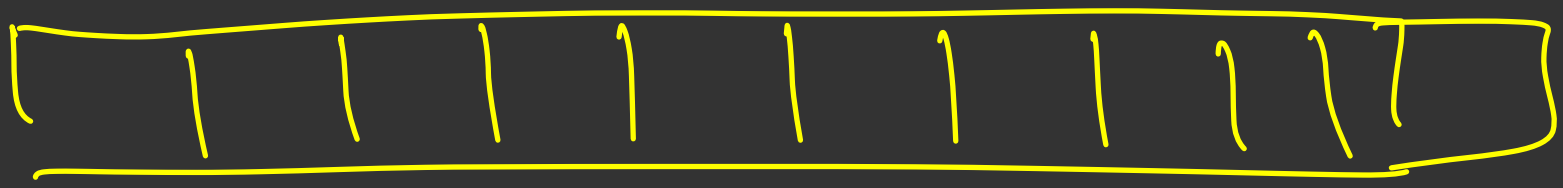
0 2nd index contains 11 and it is not the majority element.

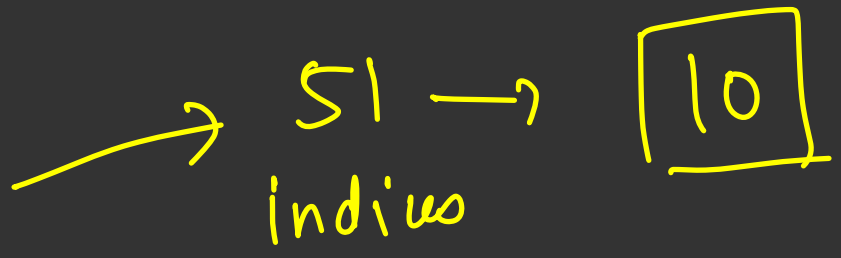
? 1

index = 1 yes

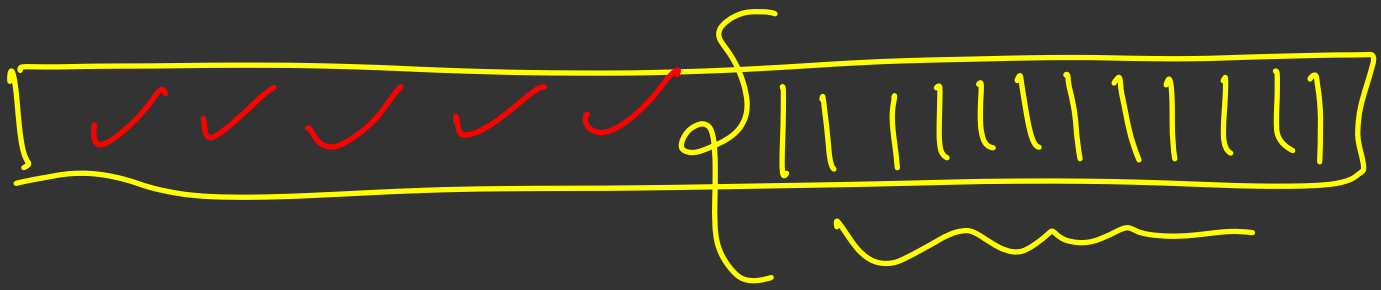
1 1st index contains 10 and it is the majority element

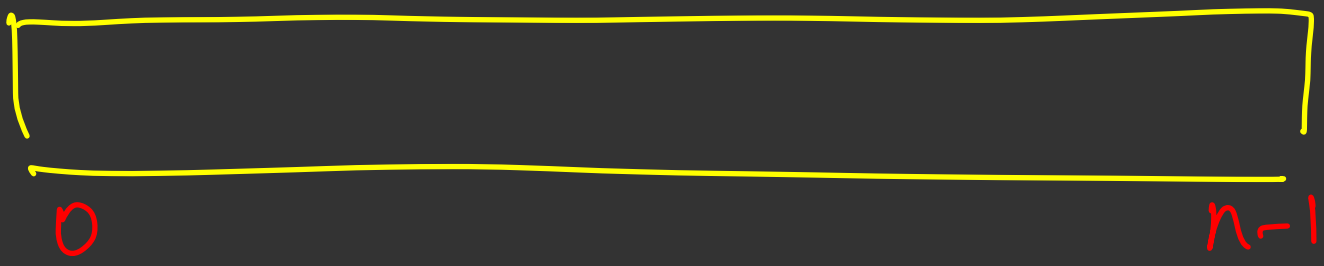
O(n) No





10^{18}





M ⟹ it occur on more than $n/2$ indices

0 1 2 3 4 5

Guess a random index

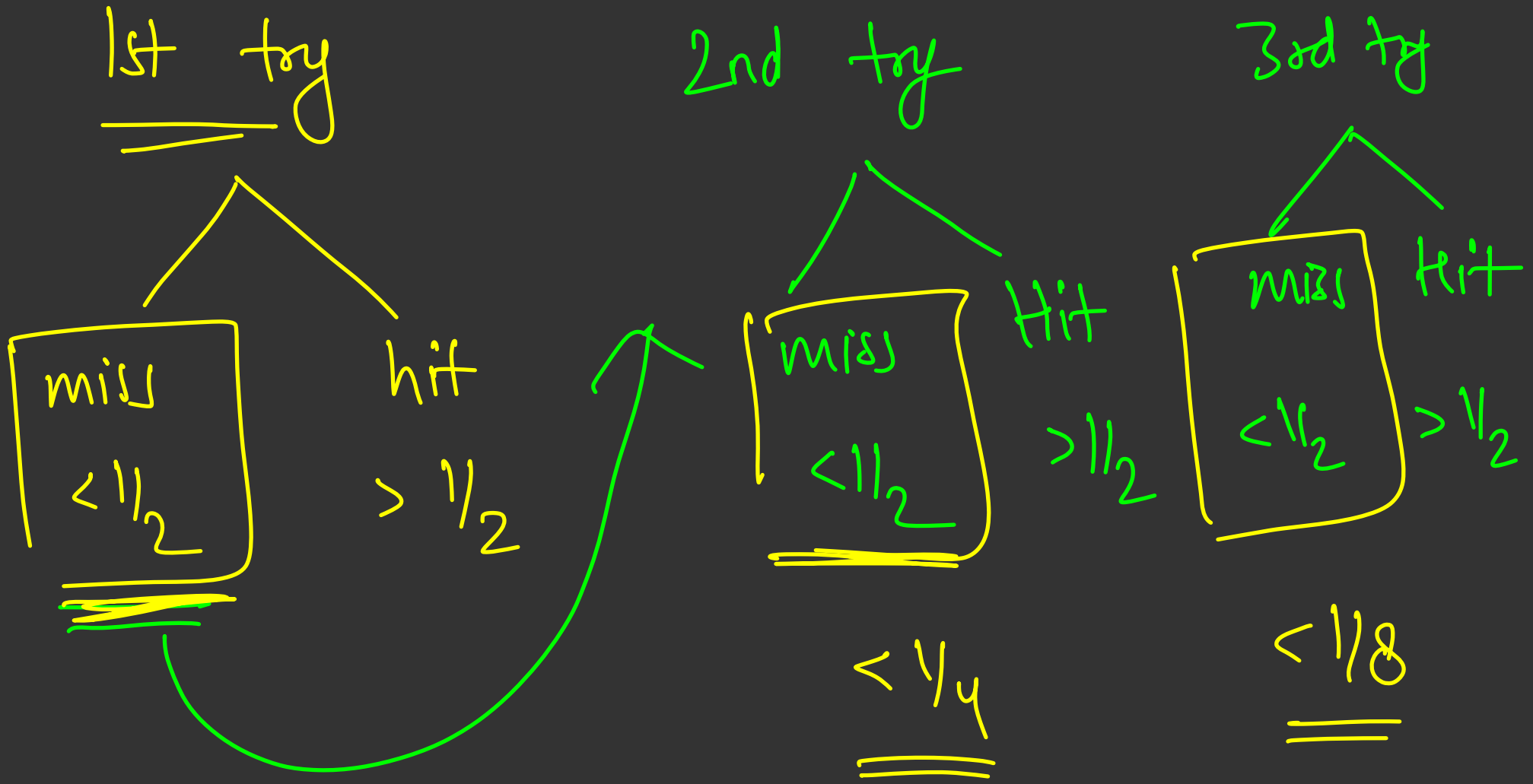
P(Hit) what is the probability that majority element will be there

$\longrightarrow > \frac{n/2}{n} \Rightarrow > 1/2$ $\frac{n/2}{n}$

$P(\text{miss}) = 1 - P(\text{Hit})$

$\Longrightarrow$

$< 1 - 1/2 = < 1/2$


1st try
miss
$< 1/2$
hit
$> 1/2$
2nd try
miss
$< 1/2$
Hit
$> 1/2$
$< 1/4$
3rd try
miss
$< 1/2$
Hit
$> 1/2$
$< 1/8$

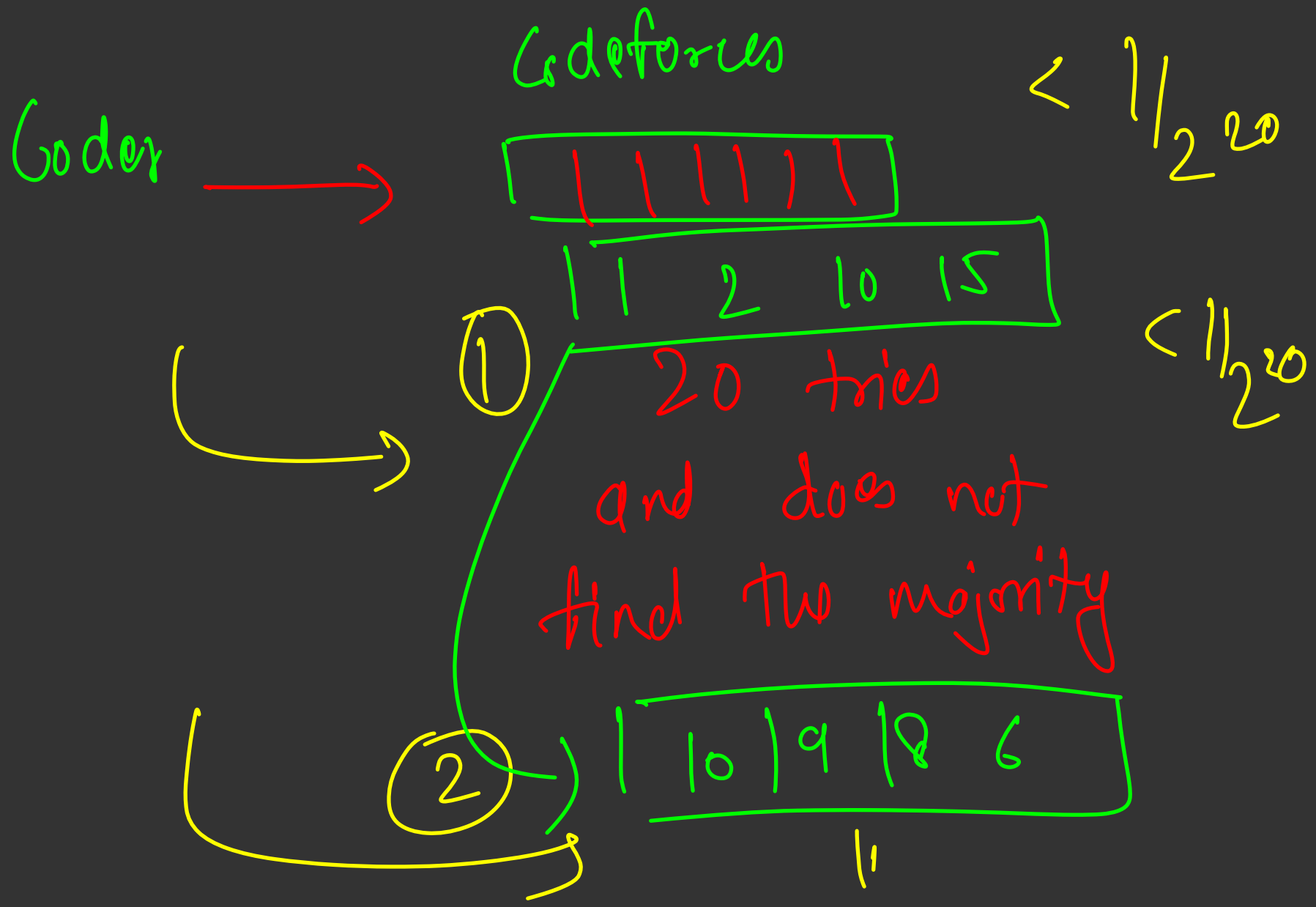
Probability of not finding majority element after k tries

randomization

$$< \frac{1}{2^k}$$

$k = 10$ $\quad < \frac{1}{2^{10}} = < \frac{1}{1000} < 0.001$

$k = 20$ $\quad < \frac{1}{2^{20}} = < \frac{1}{10^6} < 0.000001$


Codeforces
Coder
< 1/2^20
1 1 1 1 1
1 1 2 10 15
1
< 1/2^20
20 tries
and does not
find the majority
2
1 10 19 18 6
1

Hidden Array = [9, 10, 9, 9, 9, 10, 11, 10, 9, 9]

Your query	? 2	10 is not majority element
Computer's Answer	0	
Your query	? 8	11 is not majority element
Computer's Answer	0	
Your query	? 1	9 is majority element
Computer's Answer	1	
Printing the Final Answer	! 1	

Remember these 2 ideas and you will solve almost all Interactive Problems:

- Binary Search / ~~Ternary Search~~
- Randomization - Non deterministic, but once you solve a lot of problems using this, you will realise the power of this thing. Really easy to code and gives amazing results.

Side Note: Practice a lot of Constructive algo problems to improve your thinking for interactive problems.