



PROBLEM SOLVING

LeetCode 172. Factorial Trailing Zeroes ~ Intuition & Solution



NEXT SWE

9 JUL 2020 · 2 MIN READ

<https://leetcode.com/problems/factorial-trailing-zeroes/>

Based on this intuition, try solving on our own before looking at the code.

How can we tell from a number, how many trailing zeroes are at the end?

$10 = 10$

$1200 = 12 \times 10 \times 10$

$145000 = 145 \times 10 \times 10 \times 10$

From above example, we can clearly see number of 10's in a number gives us number of trailing zero in that number.

So, in our first brute force solution we can calculate the $n!$ and check how many 10's are in there. That will give us the solution.

But, since for any smaller value of n , $n!$ will become very very large to calculate. So, this is not a feasible solution.

What can we do?

Let's build some more intuition to see if we can find out the final hint we need, to solve this question.

What is the property of 10?

We can get 10 from 2 & 5 (2×5). Can you get 10 in any other way? No.

So, let's break it down further.

$10 = 2 \times 5$ [1(2),1(5)]

$100 = 2 \times 2 \times 5 \times 5$ [2(2),2(5)]

$120 = 2 \times 2 \times 2 \times 3 \times 5$ [3(2),1(3),1(5)]

$1200 = 2 \times 2 \times 2 \times 2 \times 3 \times 5 \times 5$ [4(2),1(3),2(5)] // 2 appears 4 times, 3 appears 1 time, 5 appears 2 times}

$50 = 2 \times 5 \times 5$ [1(2),2(5)]

From above you will see, among 5 and 2 whichever appears lowest amount is the count of our trailing zeros.

But for $n!$, we know we will have more 2's than 5.

Let's see an example:

$5! = 1 \times 2 \times 3 \times 4(2 \times 2) \times 5 = [3(2),1(5)]$

Try writing down couple more example and you will understand. So, in $25!$ how many 5 will be there?

$.... \times 5 \times 10 \times 15 \times 20 \times 25 = \times 5 \times 2 \times 5 \times 3 \times 5 \times 4 \times 5 \times 5 \times 5 = 6$, Because 25 itself has 2 5's in it.

Ok, up till this point we were still calculating $n!$.

But we know if n is bigger than 5 it will have 5 as a factor.

So, based on our intuition so far.

If $n=5$... there will be 1(5)

If $n=10$... there will be 2(5)

If $n=15$... there will be 3(5)

If $n=25$... there will be 5(5) + 1 (5)

Now all we need to do is keep dividing n by 5 and add the result to our existing result until n get to 0.

If you are still having problem here is the code snippet for the problem.

Solution[Java]:

```
public int trailingZeroes(int n) {
    int result = 0;
    while(n!=0){
        result += n/5;
        n = n/5;
    }
    return result;
}
```

MORE IN PROBLEM SOLVING

LeetCode 136. Single Number ~ Intuition & Solution

26 Jul 2020 – 1 min read

Cycle or Loop Detection in a LinkedList ~ find the start of the loop

15 Jul 2020 – 4 min read

LeetCode 442. Find All Duplicates in an Array ~ Intuition & Solution

14 Jul 2020 – 2 min read

[See all 6 posts →](#)

PROBLEM SOLVING

LeetCode 41. First Missing Positive ~ Intuition & Solution

<https://leetcode.com/problems/first-missing-positive/> Based on this intuition, try solving on our own before looking at the code. If we have an array of length n we know "smallest missing positive integer" must be in between 1 and $n+1$ (inclusive). For example:



NEXT SWE

9 JUL 2020 · 2 MIN READ

Study & Book Report : Data Structures and Algorithms in Java ~ Goodrich/Tamassia/Goldwasser

Best book on data structures and algorithms I have read so far. Checkout (Affiliate Links): Just wanted give a quick update on my regular study. I have been reading DS/Algo book by Goodrich/Tamassia for quite sometime now. This is turning out to



NEXT SWE

9 MAY 2020 · 7 MIN READ