

193

244

172. Factorial Trailing Zeroes

Notes

Description (/problems/factorial-trailing-zeroes/description/)

Hints (/problems/factorial-trailing-zeroes/hints/)

Submissions (/problems/factorial-trailing-zeroes/submissions/)

Simple C/C++ Solution (with detailed explanation)

19.0K

VIEWS

▲

Last Edit: 12 hours ago

(/haoel) haoel (/haoel) ★ 307

190

▼

The idea is:

1. The ZERO comes from 10.

2. The 10 comes from 2 x 5

3. And we need to account for all the products of 5 and 2. likes $4 \times 5 = 20$...

4. So, if we take all the numbers with 5 as a factor, we'll have way more than enough even numbers to pair with them to get factors of 10

Example One

How many multiples of 5 are between 1 and 23? There is 5, 10, 15, and 20, for four multiples of 5. Paired with 2's from the even factors, this makes for four factors of 10, so 23! has 4 zeros.

Example Two

Subscribe

Report

How many multiples of 5 are there in the numbers from 1 to 100?

because $100 \div 5 = 20$, so, there are twenty multiples of 5 between 1 and 100.

but wait, actually 25 is 5×5 , so each multiple of 25 has an extra factor of 5, e.g. $25 \times 4 = 100$, which introduces extra of zero.

So, we need know how many multiples of 25 are between 1 and 100? Since $100 \div 25 = 4$, there are four multiples of 25 between 1 and 100.

Finally, we get $20 + 4 = 24$ trailing zeroes in 100!

The above example tell us, we need care about 5, 5×5 , $5 \times 5 \times 5$, $5 \times 5 \times 5 \times 5$...

Example Three

By given number 4617.

$5^1 : 4617 \div 5 = 923.4$, so we get 923 factors of 5

$5^2 : 4617 \div 25 = 184.68$, so we get 184 additional factors of 5

$5^3 : 4617 \div 125 = 36.936$, so we get 36 additional factors of 5

$5^4 : 4617 \div 625 = 7.3872$, so we get 7 additional factors of 5

$5^5 : 4617 \div 3125 = 1.47744$, so we get 1 more factor of 5

$5^6 : 4617 \div 15625 = 0.295488$, which is less than 1, so stop here.

Then 4617! has $923 + 184 + 36 + 7 + 1 = 1151$ trailing zeroes.

C/C++ code

```

int trailingZeroes(int n) {
    int result = 0;
    for(long long i=5; n/i>0; i*=5){
        result += (n/i);
    }
    return result;
}

```

-----update-----

To avoid the integer overflow as **@localvar** mentioned below(in case of 'n >=1808548329'), the expression " i <= INT_MAX/5" is not a good way to prevent overflow, because 5^{13} is > INT_MAX/5 and it's valid.

So, if you want to use “multiply”, consider define the ‘i’ as ‘long long’ type.

Or, take the solution **@codingryan** mentioned in below answer!

Notes

Comments: **16**

Sort By ▼

Type comment here... (Markdown is supported)

Preview

Post

codingryan (/codingryan) ★ 35 🕒 Dec 29, 2014, 5:54 PM

⋮

```
int trailingZeroes(int n) {
    int sum=0;
    int tmp=0;
    while(n/5>0)
        r
```

[Read More](#)

35 ^ v Share Reply

SHOW 2 REPLIES

1337c0d3r (/1337c0d3r) ★ 2110 Dec 29, 2014, 7:09 PM

+1. Excellent analysis and very detailed thought process. Welcome to Discuss, @haoel!

2 ^ v Share Reply

akshaycj47 (/akshaycj47) ★ 12 Jan 23, 2018, 1:37 AM

Great explanation. Thanks a lot!

0 ^ v Share Reply

GoogleIsMyDream (/googleismydream) ★ 13 Oct 27, 2017, 8:51 PM

better than rank 1

0 ^ v Share Reply

JadenPan (/jadenpan) ★ 79 Oct 11, 2017, 9:47 AM

Really nice explanation, I forgot the situation of extra '5' induced by numbers like 25, appreciate it.

0 ^ v Share Reply

dongdl (/dongdl) ★ 60 Apr 7, 2017, 5:06 PM

Thank you for your explanation!

0 ^ v Share Reply

Izana (/izana) ★ 21 Jul 5, 2016, 4:29 PM

I remember I've seen your article years ago. Are you work at Alibaba right now? // off the topic.

0 ^ v Share Reply

JavaXu (/javaxu) ★ 30 Mar 24, 2016, 10:13 AM

Great explanation!! It's so easy to understand!! I just add the python version.

```
def trailingZeroes(self, n):
    # return n/5 + n/25 + n/125 + ... + n/(5^13)
    return sum([ n/(5**i) for i in range(1, 14) ])
```

[Read More](#)

Notes

0 ^ v Share Reply

sdlwixf (/sdlwixf) ★ 1 Jan 18, 2016, 2:16 AM

excellent! How do you come up with this solution?

0 ^ v Share Reply

gabrielwentw (/gabrielwentw) ★ 8 Dec 31, 2014, 11:19 AM

Thanks for your explanation!!

0 ^ v Share Reply

Notes

< 1 2 >

Copyright © 2018 LeetCode

[Contact Us \(/support/\)](/support/) | [Frequently Asked Questions \(/faq/\)](/faq/) | [Terms of Service \(/terms/\)](/terms/) | [Privacy Policy \(/privacy/\)](/privacy/)

 [United States \(/region/\)](/region/)