Smallest factorial number

Monday, January 3, 2022 9:16 PM

Smallest factorial number

Easy Accuracy: 50.44% Submissions: 11158 Points: 2

Given a number **n**. The task is to find the smallest number whose factorial contains at least n trailing zeroes.

Example 1:

Input:
n = 1
Output: 5
Explanation : 5! = 120 which has at least 1 trailing 0.
Example 2:

Input:
n = 6
Output: 25
Explanation : 25! has at least 6 trailing 0.

User Task:

Complete the function **findNum()** which takes an integer N as input parameters, and returns the answer.

Expected Time Complexity: O($\log_2 N * \log_5 N$).

Expected Time Complexity: $O(\log_2 N * \log_5 N)$.
Expected Auxiliary Space: $O(1)$.

Constraints:
$1 <= n <= 10^4$

Count trailing zeroes in factorial of a number

- Difficulty Level : Medium
- Last Updated : 28 Oct, 2021

Given an integer n, write a function that returns count of trailing zeroes in n!.

Examples :

```
Input: n = 5
Output: 1
Factorial of 5 is 120 which has one trailing 0.
Input: n = 20
Output: 4
Factorial of 20 is 2432902008176640000 which has
4 trailing zeroes.

Input: n = 100
Output: 24
```

We strongly recommend that you click here and practice it, before moving on to the solution.

1. **Approach:**
   A simple method is to first calculate factorial of n, then count trailing 0s in the result (We can count trailing 0s by repeatedly dividing the factorial by 10 till the remainder is 0).

2. The above method can cause overflow for slightly bigger numbers as the factorial of a number is a big number (See factorial of 20 given in above examples). The idea is to consider prime factors of a factorial n. A trailing zero is always produced by prime factors 2 and 5. If we can count the number of 5s and 2s, our task is done. Consider the following examples.
   n = 5: There is one 5 and 3 2s in prime factors of 5! (2 * 2 * 2 * 3 * 5). So a count of trailing 0s is 1.
   n = 11: There are two 5s and eight 2s in prime factors of 11! ($2^8 * 3^4 * 5^2 * 7$). So the count of trailing 0s is 2.

3. We can easily observe that the number of 2s in prime factors is always more than or equal to the number of 5s. So if we count 5s in prime factors, we are done. *How to count* the *total number of 5s in prime factors of n!?* A simple way is to calculate floor(n/5). For example, 7! has one 5, 10! has two 5s. It is not done yet, there is one more thing to consider. Numbers like 25, 125, etc have more than one 5. For example, if we consider 28! we get one extra 5 and the number of 0s becomes 6. Handling this is simple, first, divide n by 5 and remove all single 5s, then divide by 25 to remove extra 5s, and so on. Following is the summarized formula for counting trailing 0s.
   ```
   Trailing 0s in n! = Count of 5s in prime factors of n!
                     = floor(n/5) + floor(n/25) + floor(n/125) + ....
   ```
   Following is a program based on the above formula:

From <https://www.geeksforgeeks.org/count-trailing-zeroes-factorial-number/>

```java
// Java program to count
// trailing 0s in n!
import java.io.*;
```

```java
class GFG {
    // Function to return trailing
    // 0s in factorial of n
    static int findTrailingZeros(int n)
    {
        if (n < 0) // Negative Number Edge Case
            return -1;

        // Initialize result
        int count = 0;

        // Keep dividing n by powers
        // of 5 and update count
        for (int i = 5; n / i >= 1; i *= 5)
            count += n / i;

        return count;
    }

    // Driver Code
    public static void main(String[] args)
    {
        int n = 100;
        System.out.println("Count of trailing 0s in " + n
                           + "! is "
                           + findTrailingZeros(n));
    }
}

// This code is contributed by Pramod Kumar
```

Output :

```
Count of trailing 0s in 100! is 24
```

Time Complexity: *O(log5n)*

Auxiliary Space: *O(1)*

This article is contributed by **Rahul Jain**. Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above

From <https://www.geeksforgeeks.org/count-trailing-zeroes-factorial-number/>

Smallest number with at least n trailing zeroes in factorial

- Difficulty Level : Medium
- Last Updated : 08 Jun, 2021

Given a number **n**. The task is to find the smallest number whose factorial contains at least n trailing zeroes.

Examples :

```
Input : n = 1
```

```
Output : 5
1!, 2!, 3!, 4! does not contain trailing zero.
5! = 120, which contains one trailing zero.
Input : n = 6
Output : 25
```


Recommended: Please solve it on "***PRACTICE***" first, before moving on to the solution.

In the article for Count trailing zeroes in factorial of a number, we have discussed number of zeroes is equal to number of 5's in prime factors of x!. We have discussed below formula to count number of 5's.

```
Trailing 0s in x! = Count of 5s in prime factors of x!
                  = floor(x/5) + floor(x/25) + floor(x/125) + ....
```

Let us take few examples to observe pattern

```
5!  has 1 trailing zeroes
[All numbers from 6 to 9
 have 1 trailing zero]
10! has 2 trailing zeroes
[All numbers from 11 to 14
 have 2 trailing zeroes]

15! to 19! have 3 trailing zeroes

20! to 24! have 4 trailing zeroes

25! to 29! have 6 trailing zeroes
```

We can notice that, the maximum value whose factorial contain n trailing zeroes is 5*n.

So, to find minimum value whose factorial contains n trailing zeroes, use binary search on range from 0 to 5*n. And, find the smallest number whose factorial contains n trailing zeroes.

- C++
- Java
- Python3
- C#
- PHP
- Javascript

```java
// Java program tofind smallest number whose
// factorial contains at least n trailing
// zeroes.

class GFG
{
    // Return true if number's factorial contains
    // at least n trailing zero else false.
    static boolean check(int p, int n)
    {
        int temp = p, count = 0, f = 5;
        while (f <= temp)
        {
            count += temp / f;
            f = f * 5;
        }
        return (count >= n);
    }

    // Return smallest number whose factorial
    // contains at least n trailing zeroes
    static int findNum(int n)
    {
        // If n equal to 1, return 5.
        // since 5! = 120.
        if (n==1)
            return 5;

        // Initialising low and high for binary
        // search.
        int low = 0;
        int high = 5 * n;

        // Binary Search.
        while (low < high)
        {
            int mid = (low + high) >> 1;

            // Checking if mid's factorial
            // contains n trailing zeroes.
            if (check(mid, n))
                high = mid;
            else
                low = mid + 1;
        }

        return low;
    }

    // Driver code
    public static void main (String[] args)
    {
        int n = 6;
        System.out.println(findNum(n));
    }
}
```

```
// This code is contributed by Anant Agarwal.
```

Output :

```
25
```

From <https://www.geeksforgeeks.org/smallest-number-least-n-trailing-zeroes-factorial/>