Documentation: Calculating Trailing Zeroes in Factorials

Problem Overview

- The problem requires determining the number of trailing zeroes in the factorial of a given integer (n).

Factorial Definition:

- *The factorial of a non-negative integer (n), denoted as (n!), is the product of all positive integers less than or equal to (n):*
    - [n! = n times (n 1) times (n 2) times ldots times 3 times 2 times 1]

Trailing Zeroes:

- A trailing zero is a zero at the end of a number. For example, the number 100 has two trailing zeroes. The presence of trailing zeroes in the result of a factorial is directly related to the factors of 10 in the number. Since (10 = 2 times 5), each trailing zero in a number is the result of multiplying a factor of 2 and a factor of 5.

Key Insight

- To determine the number of trailing zeroes in (n!), we need to count the number of times 10 is a factor in the numbers from 1 to (n). Given that 10 is the product of 2 and 5, and that there are generally more factors of 2 than factors of 5 in a factorial, the number of trailing zeroes is determined by the number of times 5 is a factor in the sequence of numbers.

Mathematical Approach

To compute the number of trailing zeroes in (n!), we need to:

1. **Count the Factors of 5:**
   - Every multiple of 5 contributes at least one factor of 5.
   - Every multiple of (25 = 5^2) contributes an additional factor of 5.
   - Every multiple of (125 = 5^3) contributes yet another additional factor of 5.
   - And so on.

2. **Logarithmic Counting:**
   - To find the total count of trailing zeroes, we sum up the count of multiples of 5, multiples of 25, multiples of 125, etc., up to (n).

Steps to Solve the Problem

1. **Initialize a Counter:** Start with a counter set to zero. This counter will store the number of trailing zeroes.

2. **Iterative Division:**
   - Repeatedly divide (n) by 5, 25, 125, etc., until (n) becomes less than 5.
   - For each division, add the quotient to the counter. This quotient represents the count of numbers divisible by 5, 25, 125, etc., respectively.

3. **Return the Counter:** Once the iterative process completes, the counter will contain the number of trailing zeroes in (n!).

Example 1:

- **Input:** (n = 3)
- **Calculation:** (3! = 3 times 2 times 1 = 6)
- **Trailing zeroes:** 0
- **Output:** 0

Example 2:

- **Input:** (n = 5)
- **Calculation:** (5! = 5 times 4 times 3 times 2 times 1 = 120)
- **Trailing zeroes:** 1
- **Output:** 1

Example 3:

- **Input:** (n = 0)
- **Calculation:** (0! = 1) (by definition, 0! is 1)
- **Trailing zeroes:** 0
- **Output:** 0

Time Complexity Analysis

- The solution operates in logarithmic time complexity, (O(log n)), because each step divides (n) by 5. This ensures a fast and efficient computation even for large values of (n), such as (n = 10^4).

Space Complexity Analysis

- The solution uses constant space, (O(1)), as it only requires a few variables to store the counter and the modified values of (n) during the computation process. No additional data structures are required.

Conclusion

- This approach provides an efficient method to compute the number of trailing zeroes in the factorial of a given number (n) by focusing on the number of times 5 appears as a factor in the numbers from 1 to (n). The solution is both time and space-efficient, making it suitable for large inputs.