Problem 172: Factorial Trailing Zeroes

Problem Information

Difficulty: Medium
Acceptance Rate: 0.00%
Paid Only: No

Problem Description

Given an integer

n

, return

the number of trailing zeroes in

n!

.

Note that

n! = n * (n - 1) * (n - 2) * ... * 3 * 2 * 1

.

Example 1:

Input:

n = 3

Output:

0

Explanation:

3! = 6, no trailing zero.

Example 2:

Input:

n = 5

Output:

1

Explanation:

5! = 120, one trailing zero.

Example 3:

Input:

n = 0

Output:

0

Constraints:

0 <= n <= 10

4

Follow up:

Could you write a solution that works in logarithmic time complexity?

Code Snippets

C++:

```
class Solution {
public:
int trailingZeroes(int n) {

}
};
```

Java:

```
class Solution {
public int trailingZeroes(int n) {

}
}
```

Python3:

```
class Solution:
def trailingZeroes(self, n: int) -> int:
```

Python:

```
class Solution(object):
def trailingZeroes(self, n):
"""
:type n: int
:rtype: int
"""
```

JavaScript:

```
/**
* @param {number} n
* @return {number}
*/
var trailingZeroes = function(n) {

};
```

TypeScript:

```
function trailingZeroes(n: number): number {

};
```

C#:

```
public class Solution {
public int TrailingZeroes(int n) {

}
}
```

C:

```
int trailingZeroes(int n) {

}
```

Go:

```
func trailingZeroes(n int) int {

}
```

Kotlin:

```
class Solution {
fun trailingZeroes(n: Int): Int {

}
}
```

Swift:

```
class Solution {
func trailingZeroes(_ n: Int) -> Int {

}
}
```

Rust:

```
impl Solution {
pub fn trailing_zeroes(n: i32) -> i32 {

}
}
```

Ruby:

```
# @param {Integer} n
# @return {Integer}
def trailing_zeroes(n)

end
```

PHP:

```
class Solution {

/**
* @param Integer $n
* @return Integer
*/
function trailingZeroes($n) {

}
}
```

Dart:

```
class Solution {
int trailingZeroes(int n) {

}
}
```

Scala:

```
object Solution {
def trailingZeroes(n: Int): Int = {

}
}
```

Elixir:

```
defmodule Solution do
@spec trailing_zeroes(n :: integer) :: integer
def trailing_zeroes(n) do

end
end
```

Erlang:

```
-spec trailing_zeroes(N :: integer()) -> integer().
trailing_zeroes(N) ->
.
```

Racket:

```
(define/contract (trailing-zeroes n)
(-> exact-integer? exact-integer?)
)
```

Solutions

C++ Solution:

```
/*
* Problem: Factorial Trailing Zeroes
* Difficulty: Medium
* Tags: math
*
* Approach: Optimized algorithm based on problem constraints
* Time Complexity: O(n) to O(n^2) depending on approach
* Space Complexity: O(1) to O(n) depending on approach
*/
```

```
class Solution {
public:
int trailingZeroes(int n) {

}
};
```

Java Solution:

```
/**
* Problem: Factorial Trailing Zeroes
* Difficulty: Medium
* Tags: math
*
* Approach: Optimized algorithm based on problem constraints
* Time Complexity: O(n) to O(n^2) depending on approach
* Space Complexity: O(1) to O(n) depending on approach
*/

class Solution {
public int trailingZeroes(int n) {

}
}
```

Python3 Solution:

```
"""
Problem: Factorial Trailing Zeroes
Difficulty: Medium
Tags: math

Approach: Optimized algorithm based on problem constraints
Time Complexity: O(n) to O(n^2) depending on approach
Space Complexity: O(1) to O(n) depending on approach
"""

class Solution:
def trailingZeroes(self, n: int) -> int:
# TODO: Implement optimized solution
pass
```

Python Solution:

```
class Solution(object):
def trailingZeroes(self, n):
"""
:type n: int
:rtype: int
"""
```

JavaScript Solution:

```
/**
* Problem: Factorial Trailing Zeroes
* Difficulty: Medium
* Tags: math
*
* Approach: Optimized algorithm based on problem constraints
* Time Complexity: O(n) to O(n^2) depending on approach
* Space Complexity: O(1) to O(n) depending on approach
*/

/**
* @param {number} n
* @return {number}
*/
var trailingZeroes = function(n) {

};
```

TypeScript Solution:

```
/**
* Problem: Factorial Trailing Zeroes
* Difficulty: Medium
* Tags: math
*
* Approach: Optimized algorithm based on problem constraints
* Time Complexity: O(n) to O(n^2) depending on approach
* Space Complexity: O(1) to O(n) depending on approach
*/

function trailingZeroes(n: number): number {
};
```

C# Solution:

```
/*
* Problem: Factorial Trailing Zeroes
* Difficulty: Medium
* Tags: math
*
* Approach: Optimized algorithm based on problem constraints
* Time Complexity: O(n) to O(n^2) depending on approach
* Space Complexity: O(1) to O(n) depending on approach
*/

public class Solution {
public int TrailingZeroes(int n) {

}
}
```

C Solution:

```
/*
* Problem: Factorial Trailing Zeroes
* Difficulty: Medium
* Tags: math
*
* Approach: Optimized algorithm based on problem constraints
* Time Complexity: O(n) to O(n^2) depending on approach
* Space Complexity: O(1) to O(n) depending on approach
*/

int trailingZeroes(int n) {

}
```

Go Solution:

```
// Problem: Factorial Trailing Zeroes
// Difficulty: Medium
// Tags: math
//
// Approach: Optimized algorithm based on problem constraints
// Time Complexity: O(n) to O(n^2) depending on approach
// Space Complexity: O(1) to O(n) depending on approach

func trailingZeroes(n int) int {

}
```

Kotlin Solution:

```
class Solution {
fun trailingZeroes(n: Int): Int {

}
}
```

Swift Solution:

```
class Solution {
func trailingZeroes(_ n: Int) -> Int {

}
}
```

Rust Solution:

```
// Problem: Factorial Trailing Zeroes
// Difficulty: Medium
// Tags: math
//
// Approach: Optimized algorithm based on problem constraints
// Time Complexity: O(n) to O(n^2) depending on approach
// Space Complexity: O(1) to O(n) depending on approach

impl Solution {
pub fn trailing_zeroes(n: i32) -> i32 {

}
}
```

Ruby Solution:

```
# @param {Integer} n
# @return {Integer}
def trailing_zeroes(n)

end
```

PHP Solution:

```
class Solution {

/**
* @param Integer $n
* @return Integer
*/
function trailingZeroes($n) {

}
}
```

Dart Solution:

```
class Solution {
int trailingZeroes(int n) {

}
}
```

Scala Solution:

```
object Solution {
def trailingZeroes(n: Int): Int = {

}
}
```

Elixir Solution:

```
defmodule Solution do
@spec trailing_zeroes(n :: integer) :: integer
def trailing_zeroes(n) do
```

```


end
end
```

Erlang Solution:

```
-spec trailing_zeroes(N :: integer()) -> integer().
trailing_zeroes(N) ->
.
```

Racket Solution:

```
(define/contract (trailing-zeroes n)
(-> exact-integer? exact-integer?)
)
```