Solution Review: Nested Loop with Multiplication (Basic)

This review provides a detailed analysis of the different ways to solve the nested loop with multiplication problem

We'll cover the following

Solution

```
n = 10  # n can be anything, this is just an example
sum = 0
pie = 3.14
var = 1
while var < n:
    print(pie)
    for j in range(1, n, 2):
        sum += 1
    var *= 3
print(sum)

```

Time Complexity

The outer loop in this problem, i.e., everything under **line 5**, runs $log_3(n)$ times since `var` will first be equal to 1, then 3, then 9, $\cdots$, until it is 3^k such that $3^k \leq n$. This means that the outer loop runs a total of $log_3(n)$ times. The inner loop, on the other hand, runs a total of $log_3(n) \times \frac{n}{2}$. So,

Statement	Number of Executions
n = 10	1
sum = 0	1
pie = 3.14	1
var = 1	1
while var < n:	$log_3(n)$
print(pie)	$log_3(n)$
j in range(1,n,2):	$log_3(n) \times \frac{n}{2}$
sum++;	$log_3(n) \times \frac{n}{2}$
var *= 3	$log_3(n)$
print(sum)	1

Running Time Complexity =

$$5 + log_3(n) + log_3(n) + (log_3(n) \times \frac{n}{2}) + (log_3(n) \times \frac{n}{2}) + (n$$

$$= 5 + 3log_3(n) + 2(\frac{nlog_3(n)}{2})$$

$$= 5 + 3log_3(n) + nlog_3(n)$$

Now, to find the Big O complexity,

1. Drop the leading constants $\Rightarrow log_3(n) + nlog_3(n)$
2. Drop lower order terms $\Rightarrow nlog_3(n)$

Using $log_3(n) = \frac{log_2(n)}{log_2(3)} = \frac{log_2(n)}{1.585}$, we can turn this into $\frac{nlog_2(n)}{1.585}$. Dropping the constants, given us:

Big O Time Complexity $\Rightarrow O(nlog_2(n))$