Solution Review: Nested Loop with Multiplication (Pro)

This review provides a detailed analysis of the different ways to solve the Nested Loop with Multiplication challenge

We'll cover the following

Solution

```
n = 10  # can be anything
sum = 0
pie = 3.14
j = 1
for var in range(n):
    while j < var:
        sum += 1
        j *= 2
    print(sum)

```

The outer loop in the main function has n iterations as it iterates on the list generated from `range(n)`. If the condition `j < var` is true, the inner loop entered. However, immediately, `j` is doubled. Note that `j` is not reset to 1 in

the code. The inner while loop will run at most once for each iteration of the outer loop. Therefore, **lines 6, 7 and 8** run $O(n)$ times each. Since we are interested in an upper bound on the worst case running time, let's assume these statements run exactly n times.

Statement	Number of Executions
n = 10	1
sum = 0	1
pie = 3.14	1
for var in range(n):	n
while j < var:	n
sum+=1;	n
j *= 2;	n
print(sum)	n

Time Complexity#

Running Time Complexity = $1 + 1 + 1 + n + n + n + n + n$

$= 3 + 5n$

To find the Big O time complexity,

1. Drop the leading constants $\Rightarrow n$
2. Drop the lower order terms $\Rightarrow n$

The Big O time complexity of the above is hence, $O(n)$