

Solution Review: Big O of Nested Loop with Addition

This review provides a detailed analysis of the time complexity of the Nested Loop with Addition problem!

We'll cover the following ^

- Solution

Solution

```
1 n = 10 # n can be anything, this is just an example
2 sum = 0
3 pie = 3.14
4
5 for var in range(1, n, 3):
6     print(pie)
7     for j in range(1, n, 2):
8         sum += 1
9         print(sum)
10
```



The line `for var in range(1,n,3):` gets executed $\frac{n}{3}$ times and the `for j in range(1,n,2):` gets executed $\frac{n}{2}$ times for each iteration of the outer loop which makes it run a total of $\frac{n}{3} \times \frac{n}{2}$ which is in $O(n^2)$.

Study the following slides for a more detailed line-by-line analysis of the calculation of the running time complexity.

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

Running Time Complexity

0



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

Running Time Complexity

1

Initializing a variable is a basic operation that costs one unit

2 of 19

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

Running Time Complexity

2

variable initialization again

3 of 19



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

Running Time Complexity

3

Initializing `pie`

4 of 19

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

Running Time Complexity

$3+n/3$

The range function generates a list BEFORE executing the for-loop. The list in this case is of all integers from 1 to n, in steps of 3. So the running time complexity of range here is $n/3$

5 of 19



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

Running Time Complexity
 $3 + n/3 + n/3$

Each time, `var` is initialized to an element of the list generated by `range()` so it takes one unit of time at every iteration.
Hence, a total of $n/3$ iterations occur which means a running time complexity of $n/3$

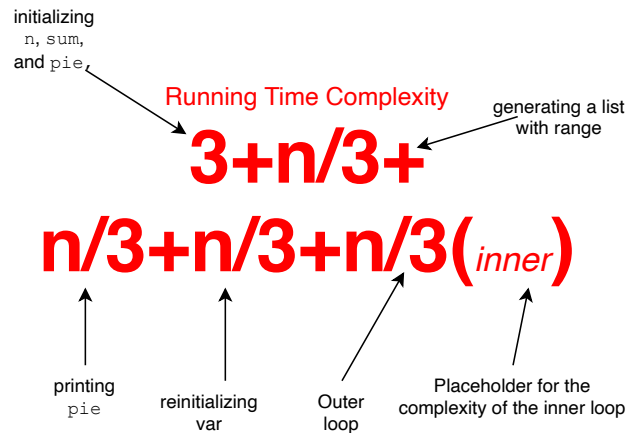
6 of 19

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

Running Time Complexity
 $3 + n/3 + n/3$
 $+ n/3$

`print(pie)` executes $n/3$ times

7 of 19



Lets now calculate the time complexity of the inner loop. Here is a recap of what we have so far. The inner loop will execute $n/3$ times so we have left a placeholder.

8 of 19

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

Running Time Complexity
inner

Lets now calculate the running time complexity of the inner loop, it will run once for every iteration of the outer loop. We've removed the rest of the running time complexity for now.

9 of 19



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

Running Time Complexity

$n/2$

range function

10 of 19

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

Running Time Complexity

$n/2+n/2$

Reinitializing j at every iteration

11 of 19



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

Running Time Complexity

$n/2 + n/2 + n$

sum is incremented and reassigned at every iteration of the loop which is each takes 2 units of time. Since this happens $n/2$ times, the total time complexity of this comes out to $(n/2) \times 2 = n$

12 of 19

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

Running Time Complexity

$n/2 + n/2$
 $+ n + n/2$

sum gets printed at every iteration of the loop which is $n/2$ times

13 of 19



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

Running Time Complexity

$5n/2$

total running time complexity of the inner loop

14 of 19

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

Running Time Complexity

$3+n/3+n/3$
 $+n/3+n/3(inner)$

lets now place the time complexity of the inner loop that we just calculated into the main running time complexity

15 of 19



Running Time Complexity

3 + n/3 + n/3

+ n/3 + n/3 (5n/2)

16 of 19

Running Time Complexity

$(5n^2)/6 + n + 3$

17 of 19



$$(5n^2)/6+n+3$$

$$\Rightarrow (n^2)+n$$
 Dropping constants

$$\Rightarrow n^2$$
 Dropping lower order terms

18 of 19

$$n^2$$

Final Big O complexity!

19 of 19



Hence, Big O time complexity: $O(n^2)$

← Back

Next →

Challenge 1: Big O of Nested Loop wit...

Challenge 2: Big O of Nested Loop wit...

✓ Completed

Report an Issue

Ask a Question

(https://discuss.educative.io/tag/solution-review-big-o-of-nested-loop-with-addition__introduction-to-complexity-measures__data-structures-for-coding-interviews-in-python)

