Example 2: Measuring Time Complexity

Compute time complexity of a given algorithm with nested loops.

We'll cover the following

In the previous lesson we went through a simple loop example and computed its time complexity by summing up all the basic operations.

A Nested Loop

Now we will extend the previous example and make it a little harder by adding a “nested loop” in the program. We will calculate its time complexity by following the same series of steps that we did in the previous example. Let’s take a look at the following example. It is a simple piece of code that prints the number of times the increment statement runs throughout the program. Let’s compute its time complexity.

```
1  n = 5  # n can be anything
2  sum = 0
3  for i in range(n):
4      for j in range(n):
5          sum += 1
6
7  print(sum)
8
```

Time Complexity

We will first break this program into individual operations like this:

Statement	Number of Executions
`n = 5`	1
`sum = 0`	1
`range(n)` **line 3**	1

Statement	Number of Executions
`i = 0`	1
`i = 1`	1
`i = 2`	1
...	
`i = n - 1`	1
`range(n)` **line 4**	$n \times 1$
`j = 0`	$1 \times n$
`j = 1`	$1 \times n$
`j = 2`	$1 \times n$
...	
`j = n - 1`	$1 \times n$
`sum+=1`	$(3n) \times n$
`print(sum)`	2
Total	$5n^2 + 2n + 4$

We're multiplying every statement of the inner loop with n because the outer loop runs it n times

Also, note that, while `range(n)` executes only once, it's execution cost is n, i.e., each call to `range(n)` results in `n` individual operations.

Time Complexity

$$= 1 + 1 + n + (1 + 1 + 1 + ... + 1) + n \times n + n(1 + 1 + 1 + ... + 1) + (3n) \times n + 2$$

$$\Rightarrow 2 + n + n + n^2 + n^2 + 3n^2 + 2$$

$$\Rightarrow 5n^2 + 2n + 4$$

Quick Quiz

Q

Suppose that in the above code the variable `sum` is iniatialized to zero in the inner `for` loop.

```
n = 5  # n can be anything
sum = 0
for i in range(n):
    sum = 0
    for j in range(n):
        sum += 1

print(sum)
```

What will be the time complexity for the code?

A) $5n^2 + 2n + 3$

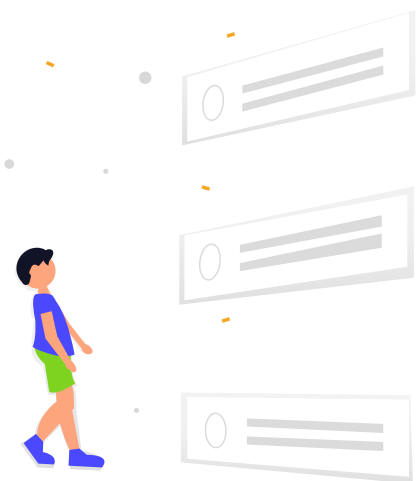
B) $5n^2 + 3n + 2$

Correct Answer

C) $5n^2 + 3n + 4$

Explanation

If we make this change, sum = 0 is run n times, whereas before it was running once. So, there's an increase of n invocations for that line.



You skipped the question. Would you like to try again?

Retake Quiz

Back

Example 1: Measuring Time Complexity

Next

Introduction to Asymptotic Analysis a...

Completed