Time Complexity

Mohammed Rizin
Unemployed

March 20, 2025

1 Time Complexity

1.1 Simple For Loop

Algorithm 1 Simple for loop

```
for i ← 0 to n − 1 do
    ....STMT....
end for
```

Time complexity: $O(n)$.

1.2 Simple Reverse Loop

Algorithm 2 Simple reverse loop

```
for i ← n downto 1 do
    ....STMT....
end for
```

Time complexity: $O(n)$.

1.3 For Loop with Step

Algorithm 3 Simple for loop with step

```
for i ← 0 to n − 1 step 2 do
    ....STMT....
end for
```

Time complexity: $O(n)$ (as $n/2$ is asymptotically $O(n)$).

1.4 Nested Loops

Algorithm 4 Nested loops

```
for i ← 0 to n − 1 do
    for j ← 0 to i − 1 do
        ....STMT....
    end for
end for
```

Time complexity: $O(n^2)$.

i	j	STMT	Total STMT	Total Time	Time Complexity
0	0	1	1	1	1
1	0	1	2	3	3
1	1	1	3	6	6
2	0	1	4	10	10
2	1	1	5	15	15
2	2	1	6	21	21

The total number of executions is $n(n+1)/2$, so the time complexity is $O(n^2)$.

1.5 Summation Loop

Algorithm 5 Summation loop

```
p ← 0
for i ← 1 while p ≤ n do
    p ← p + i
end for
```

i	p	STMT	Total STMT	Total Time	Time Complexity
1	1	1	1	1	1
2	3	1	2	3	3
3	6	1	3	6	6
4	10	1	4	10	10
5	15	1	5	15	15
6	21	1	6	21	21
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
k	$\frac{k(k+1)}{2}$	1	k	$\frac{k(k+1)}{2}$	$\frac{k(k+1)}{2}$

Assuming $p \leq n$:

$$\begin{aligned} p &= \frac{k(k+1)}{2}, \\ \frac{k(k+1)}{2} &> n, \\ k^2 &> n, \\ k &> \sqrt{n}. \end{aligned}$$

Time complexity: $O(\sqrt{n})$.