Time Complexity

Mohammed Rizin
Unemployed

March 20, 2025

1 Time Complexity

1.1 Simple For Loop

Algorithm 1 Simple for loop

for $i \leftarrow 0$ **to** $n - 1$ **do**
  ....$STMT$....
end for

Time complexity: $O(n)$.

1.2 Simple Reverse Loop

Algorithm 2 Simple reverse loop

for $i \leftarrow n$ **down to** 1 **do**
  ....$STMT$....
end for

Time complexity: $O(n)$.

1.3 For Loop with Step

Algorithm 3 Simple for loop with step

for $i \leftarrow 0$ **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 $i \leftarrow 0$ **to** $n - 1$ **do**
  for $j \leftarrow 0$ **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

$p \leftarrow 0$
for $i \leftarrow 1$ **while** $p \leq n$ **do**
  $p \leftarrow p + i$
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 $p \leq n$:

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

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

1.6 Multiplication Loop

Algorithm 6 Multiplication loop

for $i \leftarrow 1$ **while** $i \leq n$ **step** $2 \cdot i$ **do**
  ...Statement...
end for

step	i
1	2
2	4
3	8
4	16
5	32
$\vdots$	$\vdots$
k	2^k

$$\begin{aligned} \text{This stops when } i > n : \\ \text{i.e. } 2^k &> n, \\ \log_2 2^k &> \log_2 n \\ k \cdot \log_2 2 &> \log_2 n \\ k &> \log_2 n \end{aligned}$$

Time complexity: $O(\log_2 n)$.
However, If you notice log function, which gives both float and integer values, then should we take floor or ceil of the floating point value.

n = 8
1
2
4
$8 < 8$ fails !!
So in total we could execute it for 3 iterations. $\log_2 8 = 3$

n = 10
1
2
4
8
$16 < 8$ fails !!
So in total we could execute it for 4 iterations. $\log_2 10 = 3.32$ $ceil(\log_2 10) = 4$

1.7 Division Loop

Algorithm 7 For Loop with Division

for $i \leftarrow n$ **while** $i \geq n$ **step** $\frac{i}{2}$ **do**
    Statement
end for

step	i
1	n
2	$\frac{n}{2}$
3	$\frac{n}{2^2}$
4	$\frac{n}{2^3}$
5	$\frac{n}{2^4}$
$\vdots$	$\vdots$
k	$\frac{n}{2^{k-1}}$
k+1	$\frac{n}{2^k}$

$$
\begin{aligned}
\frac{n}{2^{k-1}} &< 1 \\
2^{k-1} &> n \\
k-1 &> \log_2 n \\
k &> \log_2 n - 1
\end{aligned}
$$

This algorithm is $O(\log_2 n)$