

Complexity

By: Riham Katout

Videos

1- [part1](#)

2- [part2](#)

Articles

- 1- [Knowing the complexity in competitive programming](#)
- 2- [How to Analyse Loops for Complexity Analysis of Algorithms](#)

What is the time, and space complexity of the following codes ?

```
int a = 0, b = 0;
for (i = 0; i < N; i++) {
    a = a + rand();
}
for (j = 0; j < M; j++) {
    b = b + rand();
}
```

Options:

1. $O(N * M)$ time, $O(1)$ space
2. $O(N + M)$ time, $O(N + M)$ space
3. $O(N + M)$ time, $O(1)$ space
4. $O(N * M)$ time, $O(N + M)$ space

Solution $O(N + M)$ time, $O(1)$ space

```
int a = 0;
for (i = 0; i < N; i++) {
    for (j = N; j > i; j--) {
        a = a + i + j;
    }
}
```

Options:

1. $O(N)$
2. $O(N * \log(N))$
3. $O(N * \text{Sqrt}(N))$
4. $O(N * N)$

Solution $O(N * N)$

```
int i, j, k = 0;
for (i = n / 2; i <= n; i++) {
    for (j = 2; j <= n; j = j * 2) {
        k = k + n / 2;
    }
}
```

Options:

1. $O(n)$
2. $O(N \log N)$
3. $O(n^2)$
4. $O(n^2 \log n)$

Solution $O(n \log n)$

What is the time, and space complexity of the following codes ?

```
int a = 0, i = N;
while (i > 0) {
    a += i;
    i /= 2;
}
```

Options:

1. $O(N)$
2. $O(\text{Sqrt}(N))$
3. $O(N / 2)$
4. $O(\log N)$

Solution $O(\log N)$

```
for(int i=0;i<n;i++){
    i*=k;
}
```

Options:

1. $O(n)$
2. $O(k)$
3. $O(\log_k n)$
4. $O(\log_n k)$

Solution $O(\log_k n)$

```
int value = 0;
for(int i=0;i<n;i++)
    for(int j=0;j<i;j++)
        value += 1;
```

Options:

1. n
2. $(n+1)$
3. $n(n-1)$
4. $n(n+1)$

Solution $n(n-1)$

n	Possible complexities
$n \leq 10$	$\mathcal{O}(n!)$, $\mathcal{O}(n^7)$, $\mathcal{O}(n^6)$
$n \leq 20$	$\mathcal{O}(2^n \cdot n)$, $\mathcal{O}(n^5)$
$n \leq 80$	$\mathcal{O}(n^4)$
$n \leq 400$	$\mathcal{O}(n^3)$
$n \leq 7500$	$\mathcal{O}(n^2)$
$n \leq 7 \cdot 10^4$	$\mathcal{O}(n\sqrt{n})$
$n \leq 5 \cdot 10^5$	$\mathcal{O}(n \log n)$
$n \leq 5 \cdot 10^6$	$\mathcal{O}(n)$
$n \leq 10^{18}$	$\mathcal{O}(\log^2 n)$, $\mathcal{O}(\log n)$, $\mathcal{O}(1)$