

Time and Space Complexity

Assignment Solutions



Q1. What is the time, and space complexity of the following code snippet?

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

Answer: $O(N + M)$ time, $O(1)$ space

Explanation: The first loop is $O(N)$ and the second loop is $O(M)$. Since N and M are independent variables, so we can't say which one is the leading term. Therefore Time complexity of the given problem will be $O(N+M)$. Since variables size does not depend on the size of the input, therefore Space Complexity will be constant or $O(1)$

Q2. What is the time complexity of the following code snippet?

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

Answer: $O(N*N)$

Explanation:

The above code runs total no of times
 $= N + (N - 1) + (N - 2) + \dots + 1 + 0$
 $= N * (N + 1) / 2$
 $= 1/2 * N^2 + 1/2 * N$
 $O(N^2)$ times.

Q3. What is the time complexity of the following code snippet?

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

Answer: $O(n \log n)$

Explanation: j keeps doubling till it is less than or equal to n. Several times, we can double a number till it is less than n would be $\log(n)$.

Let's take the examples here.

for $n = 16, j = 2, 4, 8, 16$

for $n = 32, j = 2, 4, 8, 16, 32$

So, j would run for $O(\log n)$ steps.

i runs for $n/2$ steps.

So, total steps = $O(n/2 * \log(n)) = O(n * \log n)$

Q4. What is the time complexity of the following code snippet?

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

Answer: $O(\log N)$

Explanation: We have to find the smallest x such that ' $(N / 2^x) < 1$ OR $2^x > N$ '
 $x = \log(N)$

Q5. What will be the time complexity of the following code snippet?

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

Answer: $O(\log_k n)$

Explanation: Because loops for the k^{n-1} times, so after taking log, it becomes $\log_k n$.