Assignment Solutions | Time and space complexity analysis - 1 | Week 8

1.Calculate the time complexity for the following code snippet.

```
int c = 0;
for(int i = n; i > 0; i /= 2) {
c++;
}
```

Solution :

O(logN), because each time n gets divided by 2 in each iteration, so in total the loop will be iterated logN to the base 2 times.

2.Calculate the time complexity for the following code snippet.

```
int c = 0;
for(int i = n; i > 1; i /= i) {
c++;
}
```

Solution :

O(1), because the loop will only be iterated once. After one iteration ‘i’ will become 1 from n and the loop will break.

3.Calculate the time complexity for the following code snippet where k is some constant (k<<n).

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

Solution :

O(n/k) ~ O(n) as k is just a constant

4.Calculate the time complexity for the following code snippet.

```
int c = 0;
for(int i = 1; i < n; i *= 2) {
c++;
}
```

Solution :

O(logN) as the loop will be iterated log2N times. We can see it like the values of ‘i’

i = 1, 2, 4, ... 2k < n

2k < n

K ~ O(logn)

5.Calculate the time complexity for the following code snippet.

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

Solution :

O(n) as note here ‘i’ is being incremented by just 1 value so the loop will be iterated n times only. Value of c is increasing but it has no effect on iterations of loop.

6.Calculate the time complexity for the following code snippet.

```
int c = 0;
for(int i = 0; i < n; i++) {
for(int j = 0; j < i; j++){
c++;
}
}
```

Solution :

O(n^2) as this is a nested loop where the jth loop is begin traversed exactly ‘i’ times and ‘i’ is being incremented just once. So the total number of times the loop will get traversed will be n*(n+1)/2 ~ O(n^2).