



# CODING BLOCKS

Code Your Way To Success

# Time Complexity

## Order Complexity Analysis

**Time Complexity** is amount of time taken by the algorithm to run as a function of the input size

**Space complexity** of an algorithm quantifies the amount of space or memory taken by an algorithm to run as a function of the length of the input.

## Time Complexity Notation

1. **Theta notation** bounds a functions from above and below, so it defines exact asymptotic behavior.
2. **Big O Notation** defines an upper bound of an algorithm, it bounds a function only from above.
3. **Omega notation** provides an asymptotic lower bound.

## Various types of time complexities which can be analyzed for the algorithm:

- **Best case time complexity:** Calculating lower bound on running time of an algorithm and is the time taken in the case that causes minimum number of operations to be executed.
- **Worst case time Complexity:** Calculating upper bound on running time of an algorithm and is the time taken in the case that causes maximum number of operations to be executed.
- **Average Time complexity Algorithm:** In average case analysis, we take all possible inputs and calculate computing time for all of the inputs. Sum all the calculated values and divide the sum by total number of inputs.

## Theoretical Analysis

- Bubble Sort
- Binary Search
- Factorial
- Polynomial Evaluation

## Your turn

- Insertion sort
- Fibonacci
- Assignment 3 and assignment 4 solutions

# Complexity Analysis Examples

```
1. for (i=0; i<=n-1; i++)  
{  
    for (j=i+1; j<=k; j++)  
    {  
        constant number of operations.  
    }  
}
```

```
2. for (i=0; i<=n-1; i++)  
{  
    for (j=i+1; j<=n; j++)  
    {  
        constant number of operations.  
    }  
}
```

```
3. for (i=0; i<=n-1; )  
{  
    for (j = 0; j<k; j++)  
    {  
        constant number of operations.  
    }  
    i= i+ j;  
}
```

```
4. for (i = n; i > 0; i /= 2){
```

```
    for (j = 0; j < i; j++){
```

```
        constant number of operations.
```

```
    }
```

```
}
```