



# DESIGN & ANALYSIS OF ALGORITHMS

ASSIGNMENT 02

**MUHAMMAD HARRIS - BCS203193**

10<sup>th</sup> October, 2022

# DESIGN & ANALYSIS OF ALGORITHMS

## QUESTION 01:

Find time and space complexity of the algorithm

### ANSWER

```
for (i = 0; i < n/2; i++)           -> n/2 + 1
{
    for (j = 1; j + n/2 <= n; j++)   -> n/2(n + 1)
    {
        for (k = 1; k <= n; k = k + 2) -> n/2(n(log n + 1))
        {
            // Statements
        }
    }
}
```

$$f(n) = n/2 + 1 + (n^2+n)/2 + (n^2\log n + n^2)/2$$

$$f(n) = O(n^2 \log n)$$

$$S(n) = O(1)$$

## QUESTION 02:

Find time and space complexity of the algorithm

### ANSWER

```
i = 1;                               -> 1
while (i < n) {                       -> log2n + 1
    j = n;                             -> log2n
    while (j > 0) {                   -> log2n(log2n + 1)
        j = j/2;                     -> log2n(log2n)
    }
    i = i*2;                           -> log2n
}
```

$$f(n) = 1 + \log_2 n + 1 + \log_2 n + \log_2 n(\log_2 n + 1) + \log_2 n(\log_2 n) + \log_2 n$$

$$f(n) = O(\log n^2)$$

$$S(n) = O(1)$$

### QUESTION 03:

Find time and space complexity of the algorithm

#### ANSWER

```
for(i = 1; i < n; i = power(i,2)) {
    // statements
}
```

->  $\log_2(\log_2 n)$

$f(n) = \log_2(\log_2 n)$   
 $f(n) = O(\log(\log n))$

$S(n) = O(1)$

### QUESTION 04:

Write any two sorting algorithms

#### ANSWER

##### 1. BUBBLE SORT

```
algorithm BubbleSort(Array, n) {
    for (i = 0; i < n; i++) {
        for (j = 0; j < n - i; j++) {
            if (Array[j] > Array[j+1]) {
                Swap(Array[j], Array[j+1]);
            }
        }
    }
}
```

Time Complexity =  $O(n^2)$   
 Space Complexity =  $O(n)$

##### 2. SELECTION SORT

```
algorithm SelectionSort(Array, n) {
    for (i = 0; i < n-1; i++) {
        Index = i;
        for (j = i+1; j < n; j++) {
            if (Array[j] < Array[Index]) {
                Index = j;
            }
        }
        Swap(Array[Index], Array[i]);
    }
}
```

Time Complexity =  $O(n^2)$   
 Space Complexity =  $O(n)$