

DESIGN & Analysis of Algorithms

ASSIGNMENT 02

**MUHAMMAD HARRIS – BCS203193**

10th 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 + (n2+n)/2 + (n2logn+n2)/2

f(n) = O(n2 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 + log2n + 1 + log2n + log2n(log2n + 1) + log2n(log2n) + log2n

f(n) = O(log n2)

S(n) = O(1)

# Question 03:

**Find time and space complexity of the algorithm**

# Answer

for(i = 1; i < n; i = power(i,2)) { -> log2(log2n)

    // statements

}

f(n) = log2(log2n)

f(n) = O(log(logn))

S(n) = O(1)

# Question 04:

**Write any two sorting algorithms**

# Answer

## 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(n2)

Space Complexity = O(n)

## Selection sort

algorithm SelectionSort(Array, n) {

    for (i = 0; i < n-1; i++) {

        Index = i;

        for (j = i+1; j < n; i++) {

            if (Array[j]<Array[Index]) {

                Index = i;

            }

        }

        Swap(Array[Index],Array[i]);

    }

}

Time Complexity = O(n2)

Space Complexity = O(n)