Date Homework # 4 Day M T W T F S

19L-1196

Design and Analysis of Algorithm

To Ms. Maryam Bashir

$$C(n) = \frac{2}{n}\left[\sum_{i=0}^{n-1} C(i)\right] + n$$

Recursive

```
int C(n)
    if n==0
        return 1   // Base case
    else
        Sum = 0
        for i=0 TO n-1 {
        Sum += C(i)   Recursive case
        }
        return (2xSum/n + n)
```

Time Complexity = $1 + 2 + \cdots n$

$= n(n+1)/2$

$\Rightarrow O(n^2)$

Date _____ Day M T W T F S

Dynamic Solution

```
int D(n)
    if n==0
        return 1
    else
        commulative = 1
        for i = 1 To n
            term = 2/n (commulative) x n
            commulative += term
        return term
```

space complexity = O(1)

time complexity = O(n)

Date _____ Day M T W T F S

Bottom up
dynamic solution

```
int BottomUp(n){
    Table[n-1]
    table[0] = 1

    for i=1 To n{

        table[i] = 2/n (table[i-1]) + n

    }
    return table[n]

}
```

Space complexity = O(n)

time complexity = O(n)