



# Assignment 02

**Fall Term 2021**

## Complexity of Recursive Functions

Code IT-466

Title Analysis of Algorithm

Due Date: **Nov 26, 2021 before 1500hr**

Degree BBIT

Batch F18-IT Major

Marks 110

### Instructions:

1. Do not forget to pray before starting to attempt the paper. Trust me it helps.  
Remember! SOMEONE is always with you (Be Relaxed), and HE is also watching you (Be Honest)
2. **Question Paper is SELF EXPLANATORY. Understanding the Question Paper is part of Solution.**
3. Nothing Beyond the Finish Line will be Evaluated. Back Side of Pages is Beyond Finish Line.
4. For Calculations etc. Use the back side of the pages.
5. Error in Question will be advantageous to Student.
6. Read the Questions carefully before attempting.
7. Solve your paper using Black/Blue Pen only.
8. Attempt All Questions in a Precise Fashion.
9. Switch Off your Cellular Phones.
10. Manage Your Time.

**GOOD LUCK**

Reg. No: \_\_\_\_\_

Name: \_\_\_\_\_

Class: IT Specialization

Date: Monday, Nov 22, 2021

Section: \_\_\_\_\_

Signature: \_\_\_\_\_

Q 01.	Q 02.	Q 03.	Q 04.	Q 05.	Q 06.	Q 07.	Q 08.	Q 09.	Q 10.	Total
10	25	15	15	10	10	10	15			110

\_\_\_\_\_  
Assistant's Signature

\_\_\_\_\_  
Examiner's Signature

**Do Not Open until YOU are Advised by the Invigilator.**

**Question No 01.****10**

Write the Time Equation for the following Recursive Functions

2 × 5

```
void doit(int n){
    if (n==1)
        return;
    for (i=0; i<n; i++)
        x = x + i;
    doit(n/2);
    doit(n/2);
}
```

```
void doit(int n){
    if (n==1)
        return;
    x++;
    doit(n/2);
    doit(n/2);
}
```

```
void doit(int n){
    if (n==1)
        return;
    for (i=0; i<n; i++)
        x = x + i;
    doit(n/2);
}
```

```
public static int power(int x, int n ){
    if (n == 0)
        return 1;
    else
        return (x * power ( x , n-1));
}
```

```
public static int sum(int n){
    if (n == 1)
        return 1;
    else
        return n + sum( n - 1);
}
```







Roll No.: \_\_\_\_\_

Name \_\_\_\_\_

A02-F18-BBIT-IT Major

Analysis of Algorithms

**Question No 03.****25**

Arrange the following function by decreasing order of growth?

**7**

$$f_1 = n \log^2 n \quad f_2 = n^{\sqrt{\log n}} \quad f_3 = 2^{\log^2 n}$$

---

---

---

---

---

---

---

---

---

---

Find the Space Complexity  $S(n)$  of the Following Iterative Algorithms, Show Working.**4**

```
Func(A, m, n)
    int X[n,m], int i,j
    for(i=1; i<=m; i++)
        for(j=1; j<=n; j++)
            X[i,j]=A[j,i]
```

---

---

---

---

---

---

---

---

---

---

```
Func()
    int X[n]
    int Y[n,n]
    for(int a=1; a<=n; a++)
        for (int b=1; b<=n; b++)
            Y[a,b]=X[a]+X[b]
```

**4**

---

---

---

---

---

---

---

---

---

---

**Question No 04.****15**Find the Space Complexity  $S(n)$  of the Following Recursive Algorithms, Show Working.**DT(int n)****5**

```
    if (n==1)
        return;
    DT(n/2);
    DT(n/2);
```

---

---

---

---

---

---

---

---

---

---

$$S(n) = 3S(n-2) + n^2$$

**5**

---

---

---

---

---

---

---

---

---

---

**Func()****5**

```
    int X[m,n,p]
    int i=1, j=1, k=1
    while(i<=m)
        j=1
        while(j<=n)
            k=1
            while(k<=p)
                Read X[i,j,k]
```

---

---

---

---

---

---

---

---









**Question No 08.**

15

Find the Time Complexity of the Algorithm with following time equation using Recursive Tree

$$T(n) = 3 T\left(\frac{n}{3}\right) + cn^3$$

## Finish Line

---

Nothing beyond this line will be evaluated