

Take Home
Mid Term Examination
Fall Term 2020

Code: IT~466
Title: Analysis of Algorithm
Max Time: 80 min

Degree: BBIT
Batch: F16-IT Major
Marks: 70

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. Sharing of resources like Stationary etc. is strictly prohibited.
5. For Calculations etc. Use the back side of the pages.
6. Error in Question will be advantageous to Student.
7. Read the Questions carefully before attempting.
8. Solve your paper using Black/Blue Pen only.
9. Attempt All Questions in a Precise Fashion.
10. This Exam is Close Books, Close Notes.
11. Switch Off your Cellular Phones.
12. Manage Your Time Wisely.

Good Luck

Roll No: _____ Name: _____
Class: IT Major Date: Wednesday November 18, 2020
Section: ☐ A ☐ B Signature: _____

Q 01.	Q 02.	Q 03.	Q 04.	Q 05.	Q 06.	Q 07.	Q 08.	Q 09.	Q 10.	Total
15	15	15	15	10						70

Invigilator's Signature

Examiner's Signature

Do Not Open until YOU are Advised by the Invigilator.

**Question No 01.****15**

Find the Time complexity of the following iterative algorithms?

4

```
int i, j, k;
for(i=1; i<=n; i*=3)
    for(j=n/2; j<=n; j/=3)
        for(k=1; k<=n3; k*=3)
            Write("IBIT");
```

Time Complexity of the following Algorithm

6

```
for (i =  $\frac{n}{2}$ ; i > 1; i -= c)
    for (j = i+1; j <= n; j *= c)
        Print("IBIT");
for(int m=1; m2 <= n; m=i+2)
    for(int j=1; j <= 2*(m-1); j++)
        Print("IBIT-PU")
    for(int k=m; k <= (10-m); k++)
        Print("AoA")
```

Find the space Complexity of following Algorithm?

5

```
M(A, int x, int y, int z)
int S1=y-x+1;
int S2=z-y;
int L1[S1+1], L2[S2+1];
for(i=0; i<S1; i++)
    L1[i]=A[x+i];
for(j=0; j<S2; j++)
    L2[j]=A[y+j+1];
L1[i]=∞; L2[j]=∞;
i=0; j=0; k=0;
while(i<=S1 && j<=S2)
    if(L1[i]<L2[j])
        A[x+k]=L1[i];
        i++;
    else
        A[x+k]=L2[j];
        j++;
    k++;
```



Roll No.: _____

Name: _____

Mid Term-F17-BBIT-IT Major

Analysis of Algorithms

Take Home**Question No 02.****Void SS(int X[], int a, int b)****If(a<b)****C=P(X, a, b)****SS(X, a, c-1)****SS(X, c+1, b)**

Find the Space Complexity of the algorithm; show the Recursion Tree for the instance with array X[1,12], Assume $c = \left\lceil \frac{a+b}{4} \right\rceil$ for every call for P(X, a, b). 3+4

Space Equation:	
Space Complexity:	

Find the Time Complexity of the above algorithm when $c = \left\lceil \frac{a+b}{4} \right\rceil$ and $T(n) = n$ for P(X, a, b). 2+6

Time Equation: _____

Recursion Tree: _____



Roll No.: _____

Name: _____

Mid Term-F17-BBIT-IT Major

Analysis of Algorithms

Take Home

Question No 04.

15

Formally State the Master Theorem

5

Find Time Complexity using Master Theorem

$$T(n) = 31 T\left(\frac{n}{3}\right) + n^2 \frac{1}{(\log n)^4}$$

3

$$T(n) = 5 T\left(\frac{2n}{5}\right) + n^{-1} \log^{-2} n$$

1

$$T(n) = 4 T\left(\frac{n}{4}\right) - n \log^{-3} n$$

2

Find the Space Complexity of following code?

4

```
int F(int n)
    if (n < 2) return 1
    else return F(n-1) + F(n-2)
```



Roll No.: _____

Name: _____

Mid Term-F17-BBIT-IT Major

Analysis of Algorithms

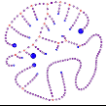
Take Home

Question No 05.

10

Compute the Time Complexity using Recursion Tree

$$T(n) = 3T\left(\frac{n}{3}\right) + n^3 \quad \forall n \geq 1$$



Roll No.: _____

Mid Term-F17-BBIT-IT Major

Name: _____

Analysis of Algorithms

Take Home

Finish Line

Nothing beyond this line will be evaluated