

Assignment-01 (50 marks)

Iterative Time Complexity + Searching + Recursive tree Construction

1. Proof That $C(n,r) = C(n-1,r-1) + C(n-1,r)$ [Here C means the combination]
[You have to give mathematical proof only] **[5 mark]**
2. Write a recursive python code to find out the value of nC_r **[5 mark]**
3. Using your recursive python code in 2, Construct a recursive tree for the value [n=5 , r=2] **[5 mark]**
4. Write a pseudocode/python code for ternary search. [Should be similar to binary search] **[5 mark]**
5. Find out the time complexity of ternary search. [The way i have shown time complexity of binary search] **[5 mark]**
6. Find out the time complexity of following Code Snippet. **[10 mark]**

```
int i,j,k,m,multi,a,b,c
for( i = n; i >= 1; i = i / 7 ){
    for( j = 1; j <= n; j = j + 3 ) {
        for( k=1; k<=40 ; k=k+1){
            multi=a*b
        }
        for( m=n ; m>=1 ; m=m-5 ){
            multi=multi*c
        }
    }
}
```

7. Find out the time complexity of following Code Snippet. [5 mark]

```
1. for i in range (1,n)
2.     j= 1
3.     while j*j < i
4.         j= j+1
```

8. Asymptotic Time complexity [2 mark]

In the primary scholarship exam in Bangladesh, four lakh ($n=4,00,000$) students take part but only the top 50 students are given an award.

Write the asymptotic time complexity to give the awards. Assume that each award is given in a constant time.

9. Asymptotic Time complexity [3 mark]

Find the time-complexity of the following task in terms of number of students.

You are given a student attendance sheet. Each student has a unique integer ID. You have to count the number of students having an even number as ID. The list is sorted but the IDs are not necessarily consecutive. So you check each ID one by one.

10. Searching: [4+1 mark]

You are given an array containing N distinct integers in a wave-like sequence. Meaning, the numbers in the beginning are in ascending order, and after a specific position, they are in descending order. For example: [1, 3, 4, 5, 9, 6, 2, -1]

You have to find the maximum number of this sequence. Can you devise an efficient algorithm such that the time complexity will be less than $O(N)$?

- Present** your solution idea as a pseudocode/ python code/ flowchart/ step-by-step instructions/ logical explanation in one-two paragraphs.
- Write** the time complexity of your algorithm.