Nikhil

Enroll. No. 92301733054

MARWADI UNIVERSITY

Faculty of Engineering and Technology

Information and Communication Technology

B.Tech

ClassTest (Written) Exam

Set-A

SEM: V

August 2025

Subject: - Design and Analysis of Algorithms (01CT0512) **Date:- 25/08/2025**

Total Marks:-30 **Time: - 50 Minutes**

Instructions:

1. Assume suitable input values if not explicitly specified in the question. Clearly mention your assumptions.
2. All questions are **compulsory.** Marks on the right side indicates total marks of the question
3. The total answer script should not exceed **4 sides of A4 sheet** (else, penalty of 2 marks may be levied for each extra sheet used).

Question: 1. Answer the following questions: [10]

1. If the time complexity of an algorithm is O(n^2), then doubling the input size will make the runtime approximately
2. Which of the following is the tightest bound for the recurrence T(n)=2T(n/2)+O(n)?
3. What is The complexity of deleting the minimum element from a min-heap of size n?
4. Which data structure gives O(1) worst-case search time?
5. In which scenario is the best case time complexity of Quick Sort is possible?
6. Which operation is time-inefficient in a singly linked list compared to arrays?
7. What is the worst case time complexity for Insertion at the end of a singly linked list, if a tail pointer is maintained?
8. What is the space complexity of an adjacency matrix representation of a graph with n vertices?
9. Which operation of a queue is performed in O(1) time in a linked list implementation but O(n) time in an array implementation (without circular array)?
10. What is the worst-case time complexity of finding the minimum element in an unsorted double-ended linked list?

Question 2 Debugging-based Questions. Mention only the line number and rectify the error [06]

Write a program to check whether a number is prime or not.

```
1. #include <iostream>
2. using namespace std;
3. int main() {
4.    int n = 11;
5.    bool isPrime = true;
6.    for(int i=2; i<n/2; i++) {
7.       if(n % i = 0) {
8.          isPrime == false;
9.          break
10.      }
11.   }
12.   if(isPrime = true)
13.      cout << "Prime" << endl;
```

Write a program to reverse a given string.

```
1. #include <iostream>
2. #include <string>
3. using namespace std;
4. int main() {
5.    string str = "hello";
6.    string rev = "";
7.    for(int i=0; i<=str.length(); i++) {
8.       rev = rev + str[i];
9.    }
10.   cout << "Reverse is:" << rev << endl
11.   return 0
12. }
```

```
else
    cout << "Not Prime" << endl;
return 0;
}
```

Question: 3. Find out the time complexity of: [03]

```
void fun(int n)
{
        for(int i=0; i<n; i++) {
            fun(n/2);
        }
}
```

Question: 4. Understand the given code and modify it to optimize the code in terms of time-complexity: [06]

```
int countOccurrences(int arr[], int n, int x) {
    int count = 0;
    for(int i=0; i<n; i++) {
        for(int j=i; j<n; j++) {
            if(arr[j] == x)
                count++;
        }
    }
    return count;
}
```

```
void printSubarrays(int arr[], int n) {
    for(int i=0; i<n; i++) {
        for(int j=i; j<n; j++) {
            for(int k=i; k<=j; k++) {
                cout << arr[k] << " ";
            }
            cout << endl;
        }
    }
}
```

Question 5: Do as directed: [05]

A browser allows navigation using the **Back** and **Forward** buttons. It uses a **singly linked list**, and moving backward requires traversing from the head.

Tasks:

1. Provide complexity for Back/Forward operations.
2. Identify inefficiency.
3. Suggest a better approach with justification.
4. Provide the complexity of the modified approach for Back/Forward operations.

~~~ ALL THE BEST ~~~

Q. No.	Marks	COs	Bloom's Level	POs
Q.1	10	CO1	Analyze	PO1, PO2, PO4
Q.2	06	CO2	Apply	PO1, PO2, PO3
Q.3	03	CO1	Analyze	PO1, PO2, PO4
Q.4	06	CO1, CO2	Evaluate	PO1, PO2, PO3
Q.5	05	CO1, CO2	Analyze	PO2, PO3, PO11