



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.



UNIVERSITY INSTITUTE OF ENGINEERING

Department of Computer Science & Engineering

(BE-CSE/IT-5th Sem)



Design and Analysis of Algorithms

Subject Code: 23CSH-301/ITH-301

Submitted to:

Faculty name: Richa Dhiman

Submitted by:

Name: Aditya Sharma

UID: 23BCS10524

Section: 607-B

Group: B



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Name: Aditya Sharma

UID: 23BCS10524

INDEX

| Ex. No | List of Experiments | Date | Conduct (MM: 12) | Viva (MM: 10) | Worksheet (MM: 8) | Total (MM: 30) | Remarks/Signature |
|--------|--|------|------------------|---------------|-------------------|----------------|-------------------|
| 1.1 | Analyze if stack is empty, is full and if elements are present then return top element in stacks using templates and also perform push and pop operation in stack. | | | | | | |
| 1.2 | Develop a program for implementation of power function and determine that complexity should be $O(\log n)$. | | | | | | |
| 1.3 | Evaluate the complexity of the developed program to find frequency of elements in a given array. | | | | | | |
| 1.4 | <ul style="list-style-type: none">i. Apply the concept of Linked list and write code to Insert and Delete an element at the beginning and end of Singly Linked List.ii. Apply the concept of Linked list and write code to Insert and Delete an element at the beginning and at end in Doubly and Circular Linked List. | | | | | | |
| 2.1 | Sort a given set of elements using the Quick sort method and determine the time required to sort the elements. Repeat the experiment for different values of n , the number of elements in the list to be | | | | | | |



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

| | | | | | | | | |
|-----|--|---|--|--|--|--|--|--|
| | | sorted. The elements can be read from a file or can be generated using the random number generator. | | | | | | |
| 2.2 | | Develop a program and analyze complexity to implement subset-sum problem using Dynamic Programming. | | | | | | |
| 2.3 | | Develop a program and analyze complexity to implement 0-1 Knapsack using Dynamic Programming. | | | | | | |
| 3.1 | | Develop a program and analyze complexity to find shortest paths in a graph with positive edge weights using Dijkstra's algorithm. | | | | | | |
| 3.2 | | Develop a program and analyze complexity to find all occurrences of a pattern P in a given string S. | | | | | | |
| 3.3 | | Lab Based Mini Project. | | | | | | |