DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

NAAC GRADE A+ Accredited University

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: MD. Shaqlain

Submitted by:

Name: Dikshay Sharma

UID: 23BCS11096
Section: KRG-1

Group: A

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 Isempty, Isfull and if elements are present then return top element in stacks using templates and also perform push and pop operation in stack.	21/07/25					
1.2	Develop a program for implementation of power function and determine that complexity should be O (log n).	01/08/25					
1.3	Evaluate the complexity of the developed program to find frequency of elements in a given array.						
1.4	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						

	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.						

Experiment No: 1.2

Student Name: Dikshay Sharma
UID: 23BCS11096
Branch: CSE
Section/Group: Krg-1A
Semester: 5th
Date of Performance: 01/08/25
Subject Name: Design analysis and algorithm
Subject Code: 23CSH-301

Aim:

Code implements power function in O(logn) time complexity.

Procedure:

1. Input: base num, exponent n.
2. If n == 0 → return 1.
3. If n < 0:
   - Set num = 1 / num.
   - Set n = -n.
4. Initialize result = 1.
5. While n > 0:
   - If n is odd → result = result * num.
   - Update num = num * num.
   - Update n = n / 2.
6. Return result.

Code:

```
#include <bits/stdc++.h>
using namespace std;

int power(int num, int n) {
    int nums = 1;
    while(n > 0) {
        if(n % 2 == 1) {
            nums *= num;
        }
        num *= num;
```

```
        n /= 2;
    }
    return nums;
}

int main() {
    int num, n;
    cin >> num >> n;
    cout << power(num, n);
    return 0;
}
```

Output:

```
50
2
2500
e:\COLLEGE\SEM 5\DAA-Lab\U1>cd "e:\COLL
\SEM 5\DAA-Lab\U1\"1.2
2
1
2
e:\COLLEGE\SEM 5\DAA-Lab\U1>
```

Time Complexity:

Time Complexity:
- The main loop runs while n > 0.
- In each iteration, n is divided by 2 (n /= 2), effectively halving n.
- Therefore, the number of iterations is approximately $\log_2(n)$.
- Each iteration performs a constant amount of work (multiplication and modulus check).
- Overall, the time complexity is O (log n).

Learning outcomes:

1. Exponentiation by Squaring
2. Bitwise Exponentiation Intuition
3. Avoiding Overflow
4. No STL Dependency Here