Experiment No: 7

Name: Aniket Balendra Tiwari

Roll No: 21143285

Program:

Optimal Binary Search Tree

```
#include <bits/stdc++.h>
using namespace std;
int sum(int freq[], int i, int j);
int optimalSearchTree(int keys[], int freq[], int n)
{
    int cost[n][n];
    for (int i = 0; i < n; i++)
        cost[i][i] = freq[i];
    for (int L = 2; L <= n; L++)
    {
        for (int i = 0; i <= n - L + 1; i++)
        {
            int j = i + L - 1;
            cost[i][j] = INT_MAX;
            int off_set_sum = sum(freq, i, j);
            for (int r = i; r <= j; r++)
            {
                int c = ((r > i) ? cost[i][r - 1] : 0) +  ((r < j) ? cost[r + 1][j] : 0) +  off_set_sum;
                if (c < cost[i][j])
                    cost[i][j] = c;
            }
        }
    }
    return cost[0][n - 1];
```

```
}
int sum(int freq[], int i, int j)
{
    int s = 0;
    for (int k = i; k <= j; k++)
        s += freq[k];
    return s;
}
int main()
{
    cout << "Name : Aniket Tiwari\n";
    cout << "Roll No : 21143285\n\n";
    int keys[] = { 10, 18, 33 };
    int freq[] = { 40, 54, 50 };
    int n = sizeof(keys) / sizeof(keys[0]);
    cout << "Cost of Optimal BST is " << optimalSearchTree(keys, freq, n);
    return 0;
}
```

Output: