**Experiment No: 7**

**OPTIMAL BINARY SEARCH TREE**

**(DYNAMIC PROGRAMMING)**

Aim: Write a program to implement OBST using dynamic programming.

Theory:

Dynamic Programming is the most powerful design technique for solving optimization problems.

Divide & Conquer algorithm partition the problem into disjoint sub problems solve the sub problems recursively and then combine their solution to solve the original problems. Dynamic Programming is used when the sub problems are not independent, e.g. when they share the same sub problems. In this case, divide and conquer may do more work than necessary, because it solves the same sub problem multiple times.

Dynamic Programming solves each sub problems just once and stores the result in a table so that it can be repeatedly retrieved if needed again.

Optimal Binary search tree

An optimal binary search tree is a binary search tree for which the nodes are arranged on levels such that the tree cost is minimum.

In OBST, a frequently used key appears near the root while a rarely used key appears far from the root. It is a weight-balanced binary tree which provides the smallest possible search time (or expected search time) for a given sequence of accesses (or access probabilities). Pi is the probability of a search being done for element ki.

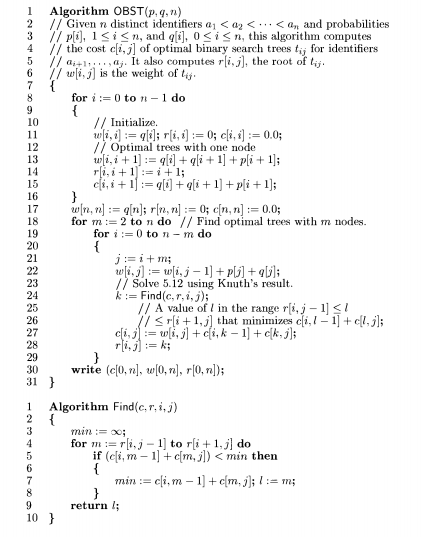
Steps:

To find the Optimal BST,

1. Generate each possible BST for the keys

2. Calculate the weighted path length.

3. Keep the tree with the smallest weighted path length



Program

Conclusion: OBST using dynamic programming was studied and implemented successfully.