OPTIMAL BINARY SEARCH TREE

TIME COMPLEXITY ANALYSIS:-

1.To initialise the cost table with 0 and it has n^2 elements to fill .so it takes o(n^2).

2. We fill the table using dynamic programming. For each subproblem size “n” we iterate through all possible subproblems of size “n”. For each subproblem we calculate the cost of constructing an optimal BST at each key in the given range. This calculation takes O(n) time. Since there are O(n^2) subproblems and each subproblem takes O(n) time to calculate .The total time complexity is O(n^3).

3.After both the steps the algorithm takes maximum time complexity that is o(n^3).

Finally the complexity of Optimal Binary Search Tree is o(n^3) where n is the number of keys. This is one of the best method to find the key in optimal way.