Assignment 3: CS21003 Algorithms 1

Prof. Partha Pratim Chakrabarti and Palash Dey
Indian Institute of Technology, Kharagpur

Submit by 11:59 PM of March 22, 20201

1. Let $a_1, \ldots, a_n$ be n distinct integers.
   (a) How many different binary search trees are possible with these n integers? Compute the average depth of these binary search trees.
   (b) How many different binary trees are possible with these n integers? Compute the average depth of these binary trees.

2. We are solving a minimization problem for a value COST, which is a positive integer. The Recursive Definition produces a binary tree structure with no identical sub-problems. The base conditions produce leaf nodes of the tree. These are solution nodes with integer COST value or dead-end nodes whose cost value is infinity. Each non-leaf sub-problem also has an integer COST value of the partial cost of that sub-problem. The costs of children of a node are always greater than the cost of the parent node. Leaf nodes can be at any depth, however the maximum depth of any leaf node is N. You are to develop an algorithm that efficiently finds the minimum COST solution and the path from start to that goal node using no more than $\mathcal{O}(N)$ space complexity. For the submission you are required to do the following:
   (a) Present the algorithm in details including the pseudo-code and data structures to be used.
   (b) Show the working on two examples of non-trivial trees, one with $N = 4$ (where all leaf nodes are at same depth) and another with $N = 5$ (where leaf nodes can be at various depths)
   (c) Prove that your algorithm correctly finds the optimal solution
   (d) Prove that your algorithm does not use more than $\mathcal{O}(N)$ space
   (e) How many solutions will your algorithm check in the worst case before finding the optimal solution?
   (f) Analyze the time complexity of your proposed algorithm.