28. Lower Bound on BST Construction:

(a) Given a Binary Search Tree (BST) holding n keys, give an efficient algorithm to print those keys in sorted order. What is the running time of the algorithm?

(b) Within the decision tree model derive a lower bound on the BST construction problem, i.e., given a set of n keys in no particular order, construct a BST that holds those n keys.

2. Coin Change making: For each of the following coin denomination systems either argue that the greedy algorithm always yields an optimum solution for any given amount, or give a counter-example:

(a) Coins c^0, c^1, c^2, …, c^{n-1}, where c is an integer > 1.

(b) Coins 1, 7, 13, 19, 61.

(c) Coins 1, 7, 14, 20, 61.

11. Small decision tree:

a) Show that every comparison based (i.e., decision tree) algorithm that sorts 5 elements, makes at least 7 comparisons in the worst-case.

b) Give a comparison based (i.e., decision tree) algorithm that sorts 5 elements using at most 7 comparisons in the worst case.

We have n balls, each with weight at most 1. More specifically, the input is an array of weights $W[1..n]$, where $W[i]$ is the weight of the i^{th} ball, $0 \le W[i] \le 1$, $i = 1..n$.
The problem is to put these balls in a **minimum** number of boxes so that:

i. each box contains no more than two balls, and
ii. the total weight of the balls placed in each box is ≤ 1.

a) [5%] Show an optimum solution for the following instance:
W = [0.36, 0.45, 0.91, 0.62, 0.53, 0.05, 0.82, 0.35].

b) [25%] Design and analyze an efficient greedy algorithm for this problem.

[Prove the correctness of your algorithm by the greedy loop invariant method, and analyze its worst-case running time.]