Problem 1. In this problem you will determine the exact number of comparisons needed to sort four elements. (Your upper and lower bounds should be the same.)

(a) Give a lower bound on the number of comparisons needed to sort four elements. Justify your answer. (Use a Comparison-Based Decision Tree.)

(b) Give an upper bound on the number of comparisons needed to sort four elements. Justify your answer. (Give an algorithm.)

Problem 2. Dr. X has written code that, given five numbers, will output them in sorted order. You are allowed to use this code in your sorting algorithm! These operations are called X’s.

(a) Define a model of computation for using these for sorting. We will call them X-trees. (It will be similar to the definition of Comparison-Based Decision Trees.)

(b) Prove a Lemma relating the number of leaves to the height of an X-tree.

(c) Give a lower bound on the number comparisons needed to sort a list of \( N \) numbers on the X-tree model.

(d) Give an algorithm for sorting using the X operations. Your algorithm’s performance does not have to match the lower bound, but should be better than \( n \lg n + \Theta(n) \).

Problem 3. Recall that with Bucket Sort we were told that the inputs are uniformly distributed. If we are told that there will be 100 times as many inputs in \([0, 1/2]\) as in \([1/2, 1]\) then how should we adjust the algorithm to still get good performance.