Binary search finds an item in a sorted table by dividing the table in half until the item is found (or it is determined that the item is not in the table). There are two versions of binary search. One version uses a two way comparison to determine that the item (if present) is in one half of the table or the other. The other version uses a three way comparison to determine that the item is in the first part of the table, is the table item being examined, or is in the last part of the table.

This question is concerned with both the worst-case and average-case time used by these algorithms. Discuss brief appropriate assumptions for the average case analysis.

The running time depends on whether the item being looked for is or is not in the table.

1. Write eight recurrence equations for the time (worst-case and average-case, in or not in, algorithm 1 or algorithm 2).
2. Solve the two average case recurrences for looking up an item that is in the table.
3. In what sense do the two algorithms take about the same time?
4. Which algorithm is faster? Comment on the significance of the difference in the speeds. In answering this question you might need to consider the following questions:
   a. How does the time to do a three way comparison compare with the time to do a two way comparison on a real computer?
   b. How many entries are in the table?