1. Give asymptotic solutions (using “big oh” notation) for each of the following recurrence equations.

   (a) $T(n) = 6T(n/4) + O(n \log n)$
   (b) $T(n) = 2T(n/4) + O(\sqrt{n})$
   (c) $T(n) = 6T(n/3) + O(n^2)$

2. Consider the problem of searching in a sorted matrix. That is, you are given an $n \times n$ matrix $A$, where each entry is an integer. Each row of the matrix is sorted in ascending order, and each column is also sorted in ascending order. Given a value $x$, the problem is to decide whether $x$ is stored somewhere in the array (i.e., whether there is some $i$ and $j$ such that $A[i][j] = x$).

   (a) One way to solve this problem is to do binary search in each row. What is the worst-case running time of this algorithm (in terms of $n$)?

   (b) Give a divide-and-conquer algorithm for this problem. (Hint: Your algorithm needs to call itself recursively, so think carefully about the parameters required. First compare $x$ with the element in the “middle” of your array (i.e., middle row, middle column). Then . . . ?).

   (c) Write down a recurrence equation describing the running time of your algorithm from (b).

   (d) Solve your equation from (c), using the formula for solving divide-and-conquer recurrences discussed in class.

   (e) Based on your answers to the above questions, which algorithm for solving the problem is faster: binary search in each row or the divide-and-conquer algorithm from part (b)?