1. Give asymptotic upper and lower bounds for \( T(n) \) for each of the following recurrences (use the Master Method when possible). Assume that \( T(n) \) is constant for \( n \) less than or equal to 2. Make your bounds as tight as possible, and justify your answers (master method should do that for you, when it is applicable). Show your work, and state the MM case that applies as well as computed values for constants.

   (a) \( T(n) = 9T(\frac{n}{4}) + n^2 \)
   (b) \( T(n) = 2T(\frac{n}{2}) + n \log^2 n \)
   (c) \( T(n) = 4T(\frac{n}{2}) + n \)
   (d) \( T(n) = 2T(n-1) + n^2 \)
   (e) \( T(n) = T(\frac{n}{2}) + n \)
   (f) \( T(n) = T(\frac{9n}{10}) + n \)
   (g) \( T(n) = 2T(\frac{n}{4}) + \sqrt{n} \)

2. Prove or disprove the following statements:

   (a) \( n^6 \in O(2^n) \)
   (b) \( 2^n \in O(3^{2^n}) \)
   (c) \( n^{\log n} \in O(2^n) \)

3. Consider the following problem of adding entries to an \( n \) by \( n \) matrix. In the upper left quadrant, place the identity matrix. In the lower right quadrant, make every entry 2. In the other two quadrants, repeat the procedure described above recursively. Show that the number of 1 entries placed in the matrix is \( \Theta(n \log n) \). Be sure to give the recurrence and solve it. You may assume \( n \) is a power of 2.

4. The following problem, called the supersink problem, was the first counterexample to the conjecture that any nontrivial problem on an arbitrary \( n \) by \( n \) 0/1 matrix requires \( \Omega(n^2) \) time. A supersink \( s \) in a matrix is an index such that \( A[s, y] = 0 \) for all values of \( y \) and \( A[x, s] = 1 \) for all \( x \neq s \). Give an \( O(n) \) algorithm to determine whether a matrix has a supersink.

5. Suppose that you have an \( n \) by \( n \) matrix in which the rows and columns are both sorted in ascending order, and you want to determine whether a particular value \( X \) is in the matrix. Design and \( O(n) \) algorithm to solve this problem.