1. **4.3-1.** Show that the solution of \( T(n) = T(n - 1) + n \) is \( O(n^2) \)

2. **4.3-3.** We saw that the solution of \( T(n) = 2T([n/2]) + n \) is \( (n \log n) \). Show that the solution of this recurrence is also \( \Omega(n \log n) \). Conclude that the solution is \( \Theta(n \log n) \).

3. **4.4-5.** Use a recursion tree to determine a good asymptotic upper bound on the recurrence \( T(n) = T(n - 1) + T(n/2) + n \). Use the substitution method to verify your answer.

4. **4.4-8.** Use a recursion tree to give an asymptotically tight solution to the recurrence \( T(n) = T(n - a) + T(a) + cnm \) where \( a \geq 1 \) and \( c > 0 \) are constants.

5. **4.5-2.** Prof. Caesar wishes to develop a matrix multiplication algorithm that is asymptotically faster than Strassen’s algorithm. His algorithm uses the divide-and-conquer method, dividing each matrix into pieces of size \( n/4 \times n/4 \), and the divide and combine steps together will take \( \Theta(n^2) \) time. He needs to determine how many subproblems his algorithm has to create in order to beat Strassen’s algorithm. If his algorithm creates \( a \) subproblems, then the recurrence for the running time \( T(n) \) becomes \( T(n) = aT(n/4) + \Theta(n^2) \). What is the largest integer value of \( a \) for which Professor Caesar’s algorithm would be asymptotically faster than Strassen’s algorithm?

6. **4.5-4.** Can the master method be applied to the recurrence \( T(n) = 4T(n/2) + n^2 \log n \)? Why or why not? Give an asymptotic upper bound for this recurrence.

7. **4-3.** Give asymptotic upper and lower bounds. Assume \( T(n) \) constant for small \( n \). Make your bound tight and justify your answers.
   - b) \( T(n) = 3T(n/3) + n/\log n. \)
   - d) \( T(n) = 3T(n/3 - 2) + n/2. \)
   - f) \( T(n) = T(n/2) + T(n/4) + T(n/8) + n. \)
   - h) \( T(n) = T(n - 1) + \log n. \)