Reminder: \( \log n \) denotes the binary logarithm, i.e., \( \log n = \log_2 n \).

1 **Recurrences: Substitution Method**

1. Consider the following recurrence:

   \[ T(1) = 1 \text{ and } T(n) = T(n-1) + n \]

   Show that \( T(n) \in O(n^2) \) using the substitution method.

2. Consider the following recurrence:

   \[ T(1) = 1 \text{ and } T(n) = T(\lceil n/2 \rceil) + 1 \]

   Show that \( T(n) \in O(\log n) \) using the substitution method.

   *Hint:* Use the inequality \( \lceil n/2 \rceil \leq \frac{n}{\sqrt{2}} = \frac{n}{2^{1/2}} \), which holds for all \( n \geq 2 \). Use \( n = 2 \) as your base case.

2 **Search in a Sorted Matrix (difficult!)**

We are given an \( n \times n \) integer matrix \( A \) that is sorted both row- and column-wise, i.e., every row is sorted in non-decreasing order from left to right, and every column is sorted in non-decreasing order from top to bottom. Give a divide-and-conquer algorithm that answers the question:

“Given an integer \( x \), does \( A \) contain \( x \)?”

What is the runtime of your algorithm?