1 Recurrences

A recurrence is a recursive definition that describes a sequence of numbers. For example, the sequence $1, 2, 3, \ldots$ is described by:

$$T_n := \begin{cases} 1 & \text{if } n = 1 \\ T_{n-1} + 1 & \text{if } n \geq 2 \end{cases}$$

Another example is the Fibonacci sequence, which is defined as:

$$T_n := \begin{cases} 1 & \text{if } 1 \leq n \leq 2 \\ T_{n-1} + T_{n-2} & \text{if } n \geq 3 \end{cases}$$

1.1 Approaches for Solving Recurrences

By solving a recurrence, we mean finding a closed-form expression for evaluating T_n. There are general approaches that can work on many types of recurrences, however, they may not always yield a solution. For two types of recurrences we also have specific approaches that always work.

1.1.1 Guess-and-Verify

This approach involves evaluating values until a pattern emerges. It does not always work.

1. Evaluate T_n for small values of n.
2. Look for a pattern in the results (this might be hard for complicated recurrences).
3. Generalize the pattern into a *guess*.
4. *Verify* the guess by induction.

1.1.2 Plug-and-Chug

This approach involves substituting the recurrence into itself until a pattern emerges. It does not always work.

1. Substitute the recurrence into itself (*plug*).
2. Simplify the result (*chug*). But not too much as it can make it harder to spot the pattern.
3. Repeat until a pattern emerges.
4. Verify the pattern by pluggin it into the recurrence one more time.

1.1.3 Approach for Linear Recurrences