# 9 - Recursion

A procedure that directly or indirectly calls itself. It has two properties:

- must have base criteria.
- each time it must be closer to the base criteria.

**Example:**

- **Factorial:**
  - if $n = 0$, then $n! = 1$.
  - if $n > 0$, then $n! = n \times (n-1)!$.
- **Fibonacci Sequence:**
  - if $n = 0$ or $n = 1$, then $F_n = n$.
  - if $n > 0$, then $F_n = F_{n-1} + F_{n-2}$.

# Divide and Conquer Algorithm

Partitions $S$ into smaller sets such that the solution of the problem $P$ for $S$ is reduced to the solution of $P$ for one or more of the smaller sets.
$[\text{here, S is the set of input data}]$

**Example:** Binary Search, Merge Sort, Quick Sort.

## Ackermann Function

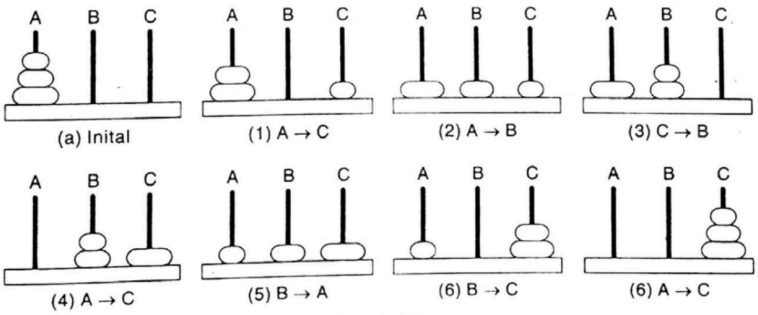
**Definition:**

- If $m = 0$, then $A(m, n) = n + 1$.
- If $m \neq 0$ but $n = 0$, then $A(m, n) = A(m-1, 1)$.
- If $m \neq 0$ and $n \neq 0$, then $A(m, n) = A(m-1, A(m, n-1))$.

**Properties:**

- Highly recursive, not primitive recursive.
- Extremely fast growth.
- Demonstrates the power of recursion.
- Very large result for small input.

## Tower of Hanoi




**Algorithm:**

1. If $N = 1$, then:
    1. Write: $BEG \rightarrow END$.
    2. Return.
2. Call $TOWER(N-1, BEG, END, AUX)$.
3. Write: $BEG \rightarrow END$.
4. Call $TOWER(N-1, AUX, BEG, END)$.
5. return.

## Is it divide-and-conquer algorithm?

Yes, the solution for $n$ disk is reduced to a solution for $n-1$ disks and a solution for $n = 1$ disk.