Examples of recursive definitions

1. arithmetic and geometric sequences
2. $a_0 = 1$, and $a_n = n * a_{n-1}$

```
int function fact(int n)
  if n==0 return 1
  else return n*fact(n-1)
```

3. Define the set $\mathcal{W}$ as follows:

   basis step: T, F, and p, where p is a propositional variable, are in $\mathcal{W}$.
   recursive step: If $E, F \in \mathcal{W}$ then each of $(\neg E)$, $(E \wedge F)$, $(E \vee F)$, $(E \rightarrow F)$, and $(E \leftrightarrow F)$ are in $\mathcal{W}$.

   Call $\mathcal{W}$ the set of **well-formed formulae** in propositional logic. Why are $\wedge pq$ and $\neg \wedge p \vee q$ not well-formed?
4. Take Σ be some set of allowable characters. Σ will contain all of the upper and lower-case letters, numbers, etc. Define λ to be the empty string, the string containing no characters. We can think of the set Σ^* of **strings over the alphabet** Σ as defined inductively:

   basis step: $\lambda \in \Sigma^*$
   recursive step: If $w \in \Sigma^*$ and $x \in \Sigma$, then $wx \in \Sigma^*$.
5. Define the **length** function for inputs from Σ^* recursively:

   basis step: $\ell(\lambda) = 0$ (length of empty string)
   recursive step: For $w \in \Sigma^*$ and $x \in \Sigma$, $\ell(wx) = \ell(w) + 1$. (Here wx is the concatenation of w followed by x.)

   Claim: For $w, z \in \Sigma^*$, $\ell(wz) = \ell(w) + \ell(z)$.
   Prove using *structural induction*, a technique for demonstrating properties that hold for elements of a recursively-defined set.

Definition 1: In **structural induction**, one shows

basis step: the result holds for all elements specified in the basis step of the recursive definition for the set.
recursive step: if the statement is true for each element used to construct new elements in the recursive step of the definition, then the result holds for these new elements.

6. Consider the Cantor middle-thirds set defined recursively as

   basis step: Start with the full interval of real numbers $C_0 = [0, 1] = \{x \mid 0 \leqslant x \leqslant 1\}$.
   recursive step: For each unbroken subinterval I still in C_n, divide I into three parts of equal length: $I = [a, b] \cup (b, c) \cup [c, d]$, and include only $[a, b] \cup [c, d]$ in C_{n+1}.

Claim: The limiting set C_∞ contains only numbers from the original interval C_0 which have ternary expansions containing only 0's and 2's.

7. We define recursively various collections of **rooted trees**. Let the set $\mathcal{R}$ be defined as follows:

    basis step: A single vertex $r \in \mathcal{R}$.
    recursive step: Suppose $T_1, T_2, \ldots T_n \in \mathcal{R}$ are disjoint having roots $r_1, \ldots, r_n$, respectively. The graph formed by taking as root a vertex r not in any of $T_1, \ldots, T_n$, and adding an edge from r to each of the vertices $r_1, \ldots, r_n$ is also in $\mathcal{R}$. Call the resulting tree $T_1 \cdot T_2 \cdots T_n$.

    Let the set $\mathcal{E}$ be defined as follows:

    basis step: The empty set is in $\mathcal{E}$.
    recursive step: Suppose T_1 and $T_2 \in \mathcal{E}$ are disjoint. The graph $T_1 \cdot T_2$ formed by taking as root a vertex r not in either T_1, nor T_2, and adding an edge from r to each of the roots of T_1, T_2 (when they are nonempty) is in $\mathcal{E}$.

    Let the set $\mathcal{F}$ be defined as follows:

    basis step: A single vertex r is in $\mathcal{F}$.
    recursive step: Suppose T_1 and $T_2 \in \mathcal{F}$ are disjoint. The graph formed by taking as root a vertex r not in either T_1, nor T_2, and adding an edge from r to each of the roots of T_1, T_2 is in $\mathcal{F}$.

    What differences between the types of trees found in $\mathcal{R}$, $\mathcal{E}$, and $\mathcal{F}$?

8. For the entries of $\mathcal{F}$ defined above, we define a **height** function recursively:

    basis step: If the tree T consists only of a root, then $h(T) = 0$.
    recursive step: For trees T_1 and $T_2 \in \mathcal{F}$, $h(T_1 \cdot T_2) = 1 + \max(h(T_1), h(T_2))$.

9. For the entries of $\mathcal{F}$ defined above, we define the set of **leaves** recursively:

    basis step: If the tree T consists only of a root r, then $L(T) = \{r\}$.
    recursive step: Given trees T_1 and $T_2 \in \mathcal{F}$, $L(T_1 \cdot T_2) = L(T_1) \cup L(T_2)$.

10. For the entries of $\mathcal{F}$ defined above, we define the set of **internal vertices** recursively:

    basis step: If the tree T consists only of a root, then $I(T) = \varnothing$.
    recursive step: Given trees T_1 and $T_2 \in \mathcal{F}$, $I(T_1 \cdot T_2) = \{r\} \cup I(T_1) \cup I(T_2)$, where r is the root of $T_1 \cdot T_2$.