**Theorem:** For any numbers $a$ and $b$, if

- $f(0) \leq a$ for $n \leq n_0$
- $f(n) \leq b + f(n-1)$ for $n > n_0$

Then $f(n) \leq a + bn$ for all $n$.

“Plug-in-and-guess”

\[
\begin{align*}
    f(n) &\leq b + f(n-1) \\
    &\leq b + b + f(n-2) \\
    &\leq b + b + b + b + f(n-3) \\
    &\leq b + b + b + b + \cdots + b + f(0) \\
    &\leq b + b + b + b + \cdots + b + f(0) \\
    &\leq b \cdot n + f(0) \\
    &\leq bn + a
\end{align*}
\]
**Theorem:** For any numbers $a$ and $b$, if
\[
\begin{align*}
    f(0) &\leq a \\
    f(n) &\leq b + f(n-1) \quad \text{for } n > 0
\end{align*}
\]
Then $f(n) \leq a + bn$ for all $n$.

**Proof:** by induction on $n$.

Base case: $n=0$. In this case, we are given $f(0) \leq a$ (Equation 1), which is what the theorem states in this case.

Induction step: Suppose $k>0$. Assume the theorem is true for $n=k-1$. That is, $f(k-1) \leq a + b \cdot (k-1)$.

Using Equation 2, \[ f(k) \leq b + f(k-1) \] Substituting, we get \[ f(k) \leq a + b + b \cdot (k-1) \] Which implies \[ f(k) \leq a + bk \]

This completes the induction step.

We have proved that $f(n) \leq a + bn$ for all $n$, i.e. Equation 3. QED
Theorem: For any numbers $a$ and $b$, if

\[ f(n) \leq a \text{ for } n \leq n_0 \]
\[ f(n) \leq bn + f(n-1) \text{ for } n > n_0 \]

Then $f(n) \leq a + bn^2$ for all $n$.

“Plug-in-and-guess”

\[ f(n) \leq bn + f(n-1) \leq bn + b\cdot(n-1) + f(n-2) \]
\[ \leq bn + b\cdot(n-1) + b\cdot(n-2) + b\cdot(n-3) + f(n-3) \]
\[ \leq bn + b\cdot(n-1) + b\cdot(n-2) + b\cdot(n-3) + \cdots + b\cdot(n_0+1) + f(n_0) \]
\[ \leq bn + bn + bn + bn + \cdots + bn + f(n_0) \]
\[ \leq bn \cdot (n-n_0) + f(n_0) \]
\[ \leq bn \cdot n + a \]
Theorem: For positive numbers $a$ and $b$ and $n_0$, if
\[
f(n) \leq a \text{ for } n \leq n_0
\]
\[
f(n) \leq bn + f(n-1) \text{ for } n > n_0
\]
Then $f(n) \leq a+bn^2$ for all $n$.

Proof: by induction on $n$.
Base case: $n \leq n_0$. In this case, we are given $f(n) \leq a$ (Equation 1), which implies $f(n) \leq a+bn^2$.
Induction step: Suppose $k > n_0$. Assume $f(n) \leq a+bn^2$ is true for $n=k-1$. That is, $f(k-1) \leq a+b(k-1)^2$.
Using Equation 2,
\[
f(k) \leq bk + f(k-1)
\]
Substituting, we get
\[
f(k) \leq a+bk+b\cdot(k-1)^2 = a+bk+bk^2-2bk+b = a+bk^2-bk+b \leq a+bk^2
\]
We have proved $f(n) \leq a+bn^2$ is true for $n=k$.
This completes the induction step, proving Equation 3. QED
Theorem: For positive numbers $a, b, c, n_0$, if
\[ f(n) \leq a \text{ for } n \leq n_0 \]
\[ f(n) \leq bn^c + f(n-1) \text{ for } n > n_0 \]
Then $f(n) \leq a + bn^{c+1}$ for all $n$.

Quiz: Prove this:
Theorem: For positive numbers $a$ and $b$ and $n_0$, if

\[ f(n) \leq a \text{ for } n \leq n_0 \]
\[ f(n) \leq bn + f(n-1) \text{ for } n > n_0 \]

Then $f(n) \leq a+bn^2$ for all $n$.

Proof: by induction on $n$.

Base case: $n \leq n_0$. In this case, we are given $f(n) \leq a$ (Equation 1), which implies $f(n) \leq a+bn^2$.

Induction step: Suppose $k > n_0$. Assume $f(n) \leq a+bn^2$ is true for $n=k-1$. That is, $f(k-1) \leq a+b(k-1)^2$.

Using Equation 2,

\[ f(k) \leq bk + f(k-1) \]

Substituting, we get

\[ f(k) \leq a+bk+b(k-1)^2 = a+bk+bk^2-2bk+b = a+bk^2-bk+b \leq a+bk^2 \]

We have proved $f(n) \leq a+bn^2$ is true for $n=k$.

This completes the induction step, proving Equation 3. QED
We say a function $f(n)$ is $O(g(n))$ if there exist nonnegative constants $n_0$ and $c$ such that, for every value of $n$ that is greater than $n_0$,

$$f(n) \leq c \cdot g(n)$$

We say a function $f(n)$ is $\Omega(g(n))$ if there exist nonnegative constants $n_0$ and $c$ such that, for every value of $n$ that is greater than $n_0$,

$$f(n) \geq c \cdot g(n)$$
**insert-in-order**

**Input:** number x, list L of numbers in nondecreasing order

**Output:** list including x and all elements of L in order

If we had this procedure, we could sort.

**insertion-sort**

Original input: (5 3 1 2 6)

Recursive input: (3 1 2 6)

Recursive output: (1 2 3 6)

Original output: (1 2 3 5 6)

**Quiz:**

```scheme
(define insertion-sort
  (lambda (L)
    (cond
      ((empty? L) empty)
      (#true (insert-in-order (car L) (insertion-sort (cdr L)))))))
```
**insert-in-order**

**Input:** number \( x \), list \( L \) of numbers in nondecreasing order

**Output:** list including \( x \) and all elements of \( L \) in order

**Original input:** \( x=5 \), \( L = (1 \ 2 \ 3 \ 6) \)

**Recursive input:** \( x=5 \), \( L=(2 \ 3 \ 6) \)

**Recursive output:** \( (2 \ 3 \ 5 \ 6) \)

**Original output:** \( (1 \ 2 \ 3 \ 5 \ 6) \)

**Original input:** \( x=1 \), \( L = (2 \ 3 \ 5 \ 6) \)

**Recursive input:** ?

**Recursive output:** ?

**Original output:** \( (1 \ 2 \ 3 \ 5 \ 6) \)

**Quiz:** Write it.