Universidad Nacional Autónoma de México

Facultad de Ciencias

Estructuras Discretas

Licenciatura en Ciencias de la Computación

Tarea 01: Naturales, inducción y recursión.

Primer Parcial

Autores:
Ramírez Mendoza Joaquín Rodrigo
Villalobos Juárez Gontran Eliut
Treviño Puebla Héctor Jerome

Agosto 2024

Tarea 01: Naturales, inducción y recursión

Ramírez Mendoza Joaquín Rodrigo
Villalobos Juárez Gontran Eliut
Treviño Puebla Héctor Jerome

September 4, 2024

1.- Demostrar por inducción que para todo natural n se cumple la siguiente igualdad:

$$\sum_{k=0}^{n} 2^k = 2^{n+1} - 1$$

Demostración por Inducción sobre n

1) Caso Base: *n=0*

$$\sum_{k=0}^{0} 2^k = 2^{0+1} - 1$$

$$2^0 = 2^1 - 1$$

$$1 = 1$$

2) Hipótesis de Inducción: Se cumple para n

$$\sum_{k=0}^{n} 2^k = 2^{n+1} - 1$$

3) Paso Inductivo: Por demostrar que se cumple para *n+1*

$$\sum_{k=0}^{n+1} 2^k = 2^{(n+1)+1} - 1$$

$$\sum_{k=0}^{n+1} 2^k = 2^{n+2} - 1$$

$$\sum_{k=0}^{n} 2^k + 2^{k+1} = 2^{n+2} - 1$$

$$\textbf{Por H.I.} \sum_{k=0}^{n} 2^k = 2^{n+1} - 1$$

$$2^{n+1} - 1 + 2^{n+1} = 2^{n+2} - 1$$

$$2^1(2^{n+1}) - 1 = 2^{n+2} - 1$$

$$2^{(n+1)+1} - 1 = 2^{n+2} - 1$$

$$2^{n+2} - 1 = 2^{n+2} - 1$$

Por lo tanto Demostramos que se cumple para toda n que:

$$\sum_{k=0}^{n} 2^k = 2^{n+1} - 1$$

2.- Demostrar que para todo natural n se cumple la igualdad:

$$\sum_{k=0}^{n} k(k+1) = \frac{n(n+1)(n+2)}{3}$$

Demostración por Inducción sobre n
1) Caso Base: *n=0*

$$\sum_{k=0}^{0} k(k+1) = \frac{0(0+1)(0+2)}{3}$$

$$0(0+1) = \frac{0(1)(2)}{3}$$

$$0(1) = \frac{0}{3}$$

$$0 = 0$$

2) Hipótesis de Inducción: Se cumple para n

$$\sum_{k=0}^{n} k(k+1) = \frac{n(n+1)(n+2)}{3}$$

3) Paso Inductivo: Por demostrar que se cumple para *n+1*

$$\sum_{k=0}^{n+1} k(k+1) = \frac{(n+1)((n+1)+1)((n+1)+2)}{3}$$

$$\sum_{k=0}^{n+1} k(k+1) = \frac{(n+1)(n+2)(n+3)}{3}$$

$$\sum_{k=0}^{n} k(k+1) + (n+1)((n+1)+1) = \frac{(n+1)(n+2)(n+3)}{3}$$

$$\sum_{k=0}^{n} k(k+1) + (n+1)(n+2) = \frac{(n+1)(n+2)(n+3)}{3}$$

Por H.I. $\sum_{k=0}^{n} k(k+1) = \frac{n(n+1)(n+2)}{3}$

$$\frac{n(n+1)(n+2)}{3} + (n+1)((n+2) = \frac{(n+1)(n+2)(n+3)}{3}$$

$$\frac{n(n+1)(n+2) + 3(n+1)(n+2)}{3} = \frac{(n+1)(n+2)(n+3)}{3}$$

$$\frac{(n+1)(n+2)(n+3)}{3} = \frac{(n+1)(n+2)(n+3)}{3}$$

Por lo tanto Demostramos que se cumple para toda n que:

$$\sum_{k=0}^{n} k(k+1) = \frac{n(n+1)(n+2)}{3}$$

3.- Para todo $n \in N$ se tiene que $2^{2n} - 1$ es múltiplo de 3.

Demostración por Inducción sobre n
1) Caso Base: *n=0*

$$2^{2(0)} - 1$$
$$2^0 - 1$$
$$1 - 1$$
$$\frac{0}{3} = 0$$
$$0$$

0 es múltiplo de 3, se cumple el Caso Base.

2) Hipótesis de Inducción: La propiedad se cumple para n

$$2^{2n} - 1 \text{ es Múltiplo de 3}$$
$$\frac{2^{2n} - 1}{3} \in N$$

3) Paso Inductivo: Por demostrar que se cumple para *n+1*

$$2^{2(n+1)} - 1$$
$$2^{2n+2} - 1$$
$$(2^{2n})(2^2) - 1$$
$$(2^{2n})(4) - 1$$
$$(4)(2^{2n}) - 1$$
$$(4)(2^{2n}) - 4 + 3$$
$$((4)(2^{2n}) - 4) + 3$$
$$(4)(2^{2n} - 1) + 3$$

Por H.I. $2^{2n} - 1$ es Múltiplo de 3

4 por un Multiplo de 3 es Múltiplo de 3 más 3 sigue siendo un número de 3

Por lo tanto Demostramos que se cumple para toda n que:

$$2^{2n} - 1 \text{ es Múltiplo de 3}$$

4.- Demostrar que para todo natural $n \geq 24$, $n = 6p + 5q$ con $p, q \in N$.

Demostración por Inducción sobre n

1) Caso Base: $n = 24$

$$24 = 6(4) + 5(0), p = 4, q = 024 = 24 + 024 = 24$$

2) Hipótesis de Inducción: Se cumple para n

$$n = 6p + 5q \text{ con } p, q \in N$$

3) Paso Inductivo: Por demostrar que se cumple para *n+1*

$$n + 1 = 6p' + 5q'$$

$$n + 1 = 6p + 5q + 1$$

$$n + 1 = 6p + 5(q - i) + 5 + 1$$

$$n + 1 = 6p + 5(q - i) + 6$$

$$n + 1 = (6p + 6) + 5(q - i)$$

$$n + 1 = 6(p + 1) + 5(q - i)$$

$$n + 1 = 6p' + 5q'$$

$$p' = p + 1, p' \in N$$

$$q' = q - 1, q' \in N$$

Por lo tanto Demostramos que se cumple para toda n que:

$$n \geq 6p + 5q,\ n \geq 24 \text{ con } p, q \in N$$

5.- Demostrar por inducción la siguiente igualdad:

$$\sum_{k=1}^{n} k^3 = \left(\sum_{k=1}^{n} k\right)^2$$

Demostración por Inducción sobre n

1) Caso Base: $n = 1$

$$\sum_{k=1}^{1} k^3 = \left(\sum_{k=1}^{1} k\right)^2$$

$$1^3 = (1)^2$$

$$1 = 1$$

2) Hipótesis de Inducción: La propiedad se cumple para n

$$\sum_{k=1}^{n} k^3 = \left(\sum_{k=1}^{n} k\right)^2$$

3) Paso Inductivo: Por demostrar que se cumple para *n+1*

$$\sum_{k=1}^{n+1} k^3 = \left(\sum_{k=1}^{n+1} k\right)^2$$

$$\sum_{k=1}^{n+1} k^3 = \left(\frac{(n+1)((n+1)+1)}{2}\right)^2$$

$$\sum_{k=1}^{n+1} k^3 = \left(\frac{(n+1)(n+2)}{2}\right)^2$$

$$\sum_{k=1}^{n} k^3 + (n+1)^3 = \left(\frac{(n+1)(n+2)}{2}\right)^2$$

Por H.I. $\sum_{k=1}^{n} k^3 = (\sum_{k=1}^{n} k)^2$

$$\left(\sum_{k=1}^{n} k\right)^2 + (n+1)^3 = \left(\frac{(n+1)(n+2)}{2}\right)^2$$

Y sabemos que: $\sum_{k=1}^{n} k = \frac{n(n+1)}{2}$

$$\left(\frac{n(n+1)}{2}\right)^2 + (n+1)^3 = \left(\frac{(n+1)(n+2)}{2}\right)^2$$

$$\frac{n^2(n+1)^2}{2^2} + (n+1)^3 = \left(\frac{(n+1)(n+2)}{2}\right)^2$$

$$\frac{n^2(n+1)^2}{4} + (n+1)^3 = \left(\frac{(n+1)(n+2)}{2}\right)^2$$

$$\frac{n^2(n+1)^2 + (4)(n+1)^3}{4} = \left(\frac{(n+1)(n+2)}{2}\right)^2$$

$$\frac{(n+1)^2(n^2 + 4(n+1))}{4} = \left(\frac{(n+1)(n+2)}{2}\right)^2$$

$$\frac{(n+1)^2(n^2+4n+4)}{4} = \left(\frac{(n+1)(n+2)}{2}\right)^2$$
$$\frac{(n+1)^2(n+2)^2}{4} = \left(\frac{(n+1)(n+2)}{2}\right)^2$$
$$\left(\frac{(n+1)(n+2)}{2}\right)^2 = \left(\frac{(n+1)(n+2)}{2}\right)^2$$

Por lo Tanto demostramos que se cumple para toda n que:

$$\sum_{k=1}^{n} k^3 = \left(\sum_{k=1}^{n} k\right)^2$$

6.- Demostrar que para los números de Fibonacci se cumle la siguiente igualdad para toda n natural:

$$F_{n+2} - 1 = \sum_{k=0}^{n} F_k$$

Demostración por Inducción sobre n

1) Caso Base: $n = 0$

$$\begin{aligned} F_{0+2} - 1 &= \sum_{k=0}^{0} F_k \\ F_2 - 1 &= F_0 \\ 1 - 1 &= 0 \\ 0 &= 0 \end{aligned}$$

2) Hipótesis de Inducción: La propiedad se cumple para n

$$F_{n+2} - 1 = \sum_{k=0}^{n} F_k$$

3) Paso Inductivo: Por demostrar que se cumple para *n+1*

$$\begin{aligned} F_{(n+1)+2} - 1 &= \sum_{k=0}^{n+1} F_k \\ F_{n+3} - 1 &= \sum_{k=0}^{n+1} F_k \\ F_{(n+3)-1} + F_{(n+3)-2} - 1 &= \sum_{k=0}^{n+1} F_k \\ F_{n+2} + F_{n+1} - 1 &= \sum_{k=0}^{n+1} F_k \\ F_{n+2} - 1 + F_{n+1} &= \sum_{k=0}^{n+1} F_k \end{aligned}$$

Por H.I. $F_{n+2} - 1 = \sum_{k=0}^{n} F_k$

$$\begin{aligned} \sum_{k=0}^{n} F_k + F_{n+1} &= \sum_{k=0}^{n+1} F_k \\ \sum_{k=0}^{n+1} F_k &= \sum_{k=0}^{n+1} F_k \end{aligned}$$

Por lo Tanto demostramos que se cumple para toda n que:

$$F_{n+2} - 1 = \sum_{k=0}^{n} F_k$$

7.- Demostrar que la suma es conmutativa; esto es, que para toda $m, n \in N$ se cumple que: $m + n = n + m$ (se debe probar que $s(m+n) = s(m)+n$, es decir, que la definición de suma también se puede hacer por la izquierda).

Demostración por inducción sobre n para $s(m + n) = s(m) + n$

a) Caso base: $n = 0$

$$s(m + 0) = s(m) \tag{1}$$
$$= s(m) + 0 \tag{2}$$

b) Hip. Ind.: Sup. que $s(m + n) = s(m) + n$

c) P.D. que se cumple para: $s(m + s(n)) = s(m) + s(n)$

$$s(m + (n + 1)) = s((m + n) + 1) \tag{3}$$
$$= s(s(m + n)) \tag{4}$$
$$\textbf{H.I} = s(s(m) + n) \tag{5}$$
$$= s(m + 1) + n \tag{6}$$
$$= (m + 1) + 1 + n \tag{7}$$
$$= m + 1 + n + 1 \tag{8}$$
$$= s(m) + s(n) \blacksquare \tag{9}$$

Por inducción sobre n para $m + n = n + m$

a) Caso base: $n = 0$

$$m + 0 = m = \tag{10}$$
$$= 0 + m \tag{11}$$

b) Hip. Ind.: Sup. que $m + n = n + m$

c) P.D. que se cumple para: $m + s(n) = s(n) + m$

$$m + s(n) = s(n) + m \tag{12}$$
$$m + s(n) = m + (n + 1) \tag{13}$$
$$= (m + n) + 1 \tag{14}$$
$$= s(m + n) \tag{15}$$
$$\textbf{H.I} = s(n + m) \tag{16}$$
$$\textbf{Por el inciso anterior} = s(n) + m \blacksquare \tag{17}$$

$$\therefore \forall n, m \in \mathbb{N}, m + n = n + m$$

8.- Demostrar que el producto es asociativo: $k \cdot (m \cdot n) = (k \cdot m) \cdot n$, para toda k,m naturales y$n \geq 1$ (HINT: se usa la propiedad distributiva).

Por inducción sobre n en la propiedad distributiva. **a) Caso base:** $n = 0$

$$\begin{aligned} (k+m)0 = 0& \\ &= 0 + 0 \\ &= k(0) + m(0) \end{aligned}$$

b) Hip. Ind.: Sup. que $(k+m)n = k \cdot n + m \cdot n$

c) P.D. que se cumple para: $(k+m)s(n) = k \cdot s(n) + m \cdot s(n)$

$$\begin{aligned} (k+m)s(n) &= s(k + m(n+1)) \\ &= (k+m)n + (k+m) \\ \textbf{Por H.I } &= k \cdot n + m \cdot n + (k+m) \\ \textbf{Por asociatividad de la suma } &= (k \cdot n + k) + (m \cdot n + m) \\ &= k(n+1) + m(n+1) \\ &= k \cdot s(n) + m \cdot s(n) \blacksquare \end{aligned}$$

Por inducción sobre n para $k \cdot (m \cdot n) = (k \cdot m) \cdot n$

a) Caso base: $n = 1$

$$\begin{aligned} k(m \cdot 1) &= k \cdot m \\ &= (k \cdot m)1 \end{aligned}$$

b) Hip. Ind.: Sup. que $k \cdot (m \cdot n) = (k \cdot m) \cdot n$

c) Paso inductivo: P.D. que se cumple para: $k \cdot (m \cdot s(n)) = (k \cdot m) \cdot s(n)$

$$\begin{aligned} k \cdot (m \cdot s(n)) &= k \cdot (m \cdot (n+1)) \\ \textbf{Por distributición con la suma } &= k \cdot (m \cdot n + m) \\ \textbf{Por distributición con la suma } &= k \cdot (m \cdot n) + k \cdot (m) \\ \textbf{Por H.I } &= (k \cdot m) \cdot n + k \cdot (m) \\ &= (k \cdot m) \cdot (n+1) \blacksquare \end{aligned}$$

$$\therefore \forall k, m, n \in \mathbb{N}, k \cdot (m \cdot n) = (k \cdot m) \cdot n$$

9.- Defínase la siguiente sucesión recursivamente:

- Base $a_1 = 1ya_2 = 3$
- Recursión $a_n = a_{n-1} + 2 \cdot a_{n-2}$

Demostrar que para toda n, a_n es impar.

a) Caso base: $a_1 = 1$

$$\begin{aligned} a_1 =& 1 \\ =& 2(0) + 1 \\ a_2 =& 3 = \qquad 2(1) + 1 \end{aligned}$$

b) Hip. Ind.: Sup. que se cumple para

$$\begin{aligned} k \leq n, a_k =& a_{k-1} + 2a_{k-2} \\ a_k =& 2m + 1 \end{aligned}$$

c) Paso inductivo: P.D. que se cumple para: $k + 1, a_{k+1} = 2m' + 1$

$$\begin{aligned} a_{k+1} =& a_{(k+1)-1} + 2 \cdot a_{(k+1)-2} \\ =& a_k + 2 \cdot a_{k-1} \\ \textbf{Por H.I} =& 2m_1 + 1 + 2(2m_2 + 1) \\ =& 2m_1 + 1 + 4m_2 + 2 \\ =& 2(m_1 + 2m_2 + 1) + 1 \\ \textbf{Notemos que } (m_1 + 2m_2 + 1) & \textbf{ es entero} \\ =& 2m' + 1 \blacksquare \end{aligned}$$

$$\therefore \textbf{existe un k entero tal que } a_n = 2k + 1$$

$$\therefore \forall a_n \in \mathbb{N}, a_n = 2k + 1 \textbf{ es impar}$$

10.- Demostrar que la concatenación de las listas tienen las siguientes propiedades:

a) Es asociativa, estos es que para toda lista l_1, l_2 y l_3 se cumple que:

$$l_1 \sqcup (l_2 \sqcup l_3) = (l_1 \sqcup l_2) \sqcup l_3$$

b) Define la funcioń de agregar elementos; esto es:

$$(a : l) = [a] \sqcup l$$

a)Demostración estructural sobre la lista l

1) Caso Base: $l = [\,]$

$$[\,] \sqcup (l_2 \sqcup l_3) = (l_2 \sqcup l_3) \tag{18}$$
$$= ([\,] \sqcup l_2) \sqcup l_3 \tag{19}$$

2) Hipótesis de Inducción: Suponemos que

$$l_1 \sqcup (l_2 \sqcup l_3) = (l_1 \sqcup l_2) \sqcup l_3$$

3) Paso Inductivo: Por demostrar que $(a : l_1) \sqcup (l_2 \sqcup l_3) = ((a : l_1) \sqcup l_2) \sqcup l_3$

$$\text{Entonces } (a : l_1) \sqcup (l_2 \sqcup l_3) = (a : (l_1 \sqcup (l_2 \sqcup l_3))) \tag{20}$$
$$\text{Por H.I } l_1 \sqcup (l_2 \sqcup l_3) = (l_1 \sqcup l_2) \sqcup l_3 \tag{21}$$
$$= (a : ((l_1 \sqcup l_2) \sqcup l_3)) \tag{22}$$
$$= (a : (l_1 \sqcup l_2)) \sqcup l_3 \tag{23}$$
$$= ((a : l_1) \sqcup l_2) \sqcup l_3 \blacksquare \tag{24}$$
$$\tag{25}$$

$\therefore$ **demostramos que para toda lista l_1, l_2 y l_3, la operación de concatenación $\sqcup$ es conmutativa**

b)Define la funcioń de agregar elementos; esto es:

$$(a : l) = [a] \sqcup l$$

Sea una lista con un elemento a que queremos combinar a la lista l y supongamos que l es una lista de elementos. Entonces nuestra función $cons(a, l)$, donde a será la cabeza y l la cola de de la lista.

Ejemplo:

$$l = [3, 8, 2, 1] \text{ y } a = 1$$
$$cons(a, l) = [1] \sqcup [3, 8, 2, 1] = [1, 3, 8, 2, 1]$$

Nuestra función sería de la forma $cons(a, l) = [a] \sqcup l$

Demostración por Inducción a la lista:

1) Caso Base:

$$cons(a, [\,]) = [a] = [a] \sqcup [\,]$$

.

2) Hipótesis de Inducción: Suponemos que

$$cons(a, l) = [a] \sqcup l$$

concatena correctamente para cualquier lista l

3) Paso Inductivo: Por demostrar que se cumple para $cons(a, (b : l)) = [a] \sqcup (b : l)$

Por H.I sabemos que la funcioń cons concatena correctamente la cadena, entonces la función regretorna el valor de$[a] \sqcup l$

$$\therefore cons(a, (b : l)) = [a] \sqcup (b : l) \tag{26}$$

12. Sea T un árbol binario con un $l = leafs(T)$ hojas y $h = depth(T) \geq 1$ de profundidad.

- a) Demostrar que el número de nodos es igual a $2l - 1$.
- b) Demostrar que $leafs(T) \leq 2^{h-1}$

Demostración por Inducción estructural sobre árboles binarios.

Para a):

$$nn(T) = 2l - 1$$

1) Caso Base: $T = tree(void, c, void)$

$$nn(T) = nn(void) + nn(void) + 1$$

$$leafs(tree(void, c, void)) = leafs(void) + leafs(void) + 1$$

$$nn(void) + nn(void) + 1 = 2(leafs(void) + leafs(void) + 1) - 1$$

$$0 + 0 + 1 = 2(0 + 0 + 1) - 1$$

$$1 = 2(1) - 1$$

$$1 = 2 - 1$$

$$1 = 1$$

2) Hipótesis de Inducción: Suponemos que se cumple para T_1 y T_2

$$nn(T_1) = 2l_1 - 1 \text{ y } nn(T_2) = 2l_2 - 1$$

3) Paso Inductivo: Por Demostrar que se cumple T

$$T = tree(T_1, c, T_2)$$

$$nn(T) = 2l_T - 1$$

$$nn(T_1) + nn(T_2) + 1 = 2l_T - 1$$

Por H.I.: $nn(T_1) = 2l_1 - 1$ y $nn(T_2) = 2l_2 - 1$

$$2l_1 - 1 + 2l_2 - 1 + 1 = 2l_T - 1$$

$$2l_1 + 2l_2 - 1 - 1 + 1 = 2l_T - 1$$

$$2l_1 + 2l_2 - 1 = 2l_T - 1$$

$$2(l_1 + 1_2) - 1 = 2l_T - 1$$

Sabemos que:

$$leafs(T) = leafs(T_1) + leafs(T_2) l_T = l_1 + l_2$$

Entonces:

$$2l_T - 1 = 2l_T - 1$$

Por lo tanto: Demostramos que se cumple para todo T que:

$$nn(T) = 2l - 1$$

Para b):

$$leafs(T) \leq 2^{h-1}$$

1) Caso Base: $T = tree(void, c, void)$

$$leafs(T) = leafs(void) + leafs(void) + 1$$

$$depth(tree(void, c, void)) = 1 + max[depth(void), depth(void)]$$

Entonces:

$$leafs(void) + leafs(void) + 1 \leq 1 + max[depth(void), depth(void)]$$

$$0 + 0 + 1 \leq 1 + max[0, 0]$$

$$1 \leq 1$$

2) Hipótesis de Inducción: Suponemos que se cumple para T_1 y T_2

$$leafs(T_1) \leq 2^{h_1-1} \text{ y } leafs(T_2) \leq 2l^{h_2-1}$$

3) Paso Inductivo: Por Demostrar que se cumple para T

$$T = tree(T_1, c, T_2)$$

- $h_T = depth(T) = 1 + max(depth(T_1), depth(T_2))$
- $h_T = 1 + max(h_1, h_2)$

$$leafs(T) \leq 2^{h_T-1}$$

$$leafs(T_1) + leafs(T_2) \leq 2^{h_T-1}$$

Por H.I.: $leafs(T_1) \leq 2^{h_1-1}$ y $leafs(T_2) \leq 2^{h_2-1}$

$$leafs(T_1) + leafs(T_2) \leq 2^{h_1-1} + 2^{h_2-1}$$

Notar que:

- $h_1 \leq max(h_1, h_2)$
- $h_2 \leq max(h_1, h_2)$

$$2^{h_1-1} + 2^{h_2-1} \leq 2^{max(h_1,h_2)-1} + 2^{max(h_1,h_2)-1}$$

$$2^{max(h_1,h_2)-1} + 2^{max(h_1,h_2)-1} = (2^1)(2^{max(h_1,h_2)-1})$$

$$(2^1)(2^{max(h_1,h_2)-1}) = 2^{max(h_1,h_2)-1+1})$$

$$2^{1+max(h_1,h_2)-1}$$

Recordemos que:

$$1 + max(h_1, h_2) = h_T$$

Entonces:

$$2^{1+max(h_1,h_2)-1} = 2^{h_T-1}$$

Por lo Tanto: Demostramos que se cumple para todo árbol T que:

$$leafs(T) \leq 2^{h-1}$$

14.- Demostrar que para todo natural a, b con $b > 0$ se cumple que la función:

$$div(a,b) = \begin{cases} (0,a) & \text{si } a < b \\ f(div(a-b,b)) & \text{si } a \geq b \end{cases} \tag{27}$$

donde $f(x,r) = (x+1,r)$ cunple que $div(a,b) = (x,r)$ con $a = x \cdot b + r$

Inducción sobre los pasos del algoritmo.

a) Caso base: En 0 pasos, tenemos que $a < b \therefore$ Regresa $x = 0$ y $r = a$. $\therefore$ queremos ver que $a = b \cdot 0 + a = 0 + a = a$ y $a = r < b$

b) Hip. Ind.: Sup. que el algoritmo en n pasos regresa c y r tal que $a = bx + r$.

c) Paso inductivo: P.D. que se cumple para $n+1$ pasos. Entonces tenemos que aplicar la función

$$f(x_n, r_n) = (x+1, r) = x_{n+1}, r_{n+1}$$

.

Donde

$$\begin{aligned} b \cdot c_{n+1} + r_{n+1} =& b(n+1) + r_n - b \\ =& b(x_n) + b + r_n - b \\ =& b(x_n) + r_n \\ \textbf{Por H.I} =& a \end{aligned}$$

$\therefore$ el algoritmo termina $\Rightarrow \forall a, b \in \mathbb{N}(b > 0)$ se cumple la función

15. Demuestra que la siguiente funcion recursiva para lista de enteros:

$$max(a:[]) = a$$

$$max(a:l) = \begin{cases} a & \text{si } a \geq max(l) \\ max(l) & \text{si } a < max(l) \end{cases}$$

Regresa siempre el Valor máximo de la lista.

Por inducción sobre los pasos del algoritmo.

1) Caso Base:

En 0 pasos → max (a:[]) = a
Regresa "a" que es el máximo de la lista que solo contiene a "a"

2) Hipótesis de Inducción:

Suponemos que el algoritmo sobre cualquier lista l en n pasos regresa $max(l)$, es decir, el valor máximo de la lista.

3) Paso Inductivo: Por Demostrar que se cumple para $(a : l)$ que la función max regresa el valor máximo. Entonces para $n + 1$ pasos, se aplica para la lista.

$$max(l') = max(a : l)$$

Debemos evaluar en dos casos:

$$max(a:l) = \begin{cases} a & \text{si } a \geq max(l) \\ max(l) & \text{si } a < max(l) \end{cases}$$

Para el Caso en que $a \geq max(l)$ se toma el valor a, Puesto que por [H.I.] sabemos que de la lista l la función $max(l)$ nos dará el valor máximo, al comparar este con a tendremos el caso en que a es mayor, por lo que podamos tomar a y también en el caso en que $a = max(l)$. Así en este caso tenemos el valor máximo de la lista y la función termina.

Para el Caso en que $max(l) > a$ también podemos afirmar que por [H.I.] se cumple tenemos el valor máximo de la lista $max(a : l)$ puesto que a no es mayor al máximo del resto de la lista, luego la función termina.

Se cumple para ambos casos.
Por lo tanto: Podemos afirmar que el algoritmo max regresa el valor máximo para toda lista l.
El algoritmo $max(l)$ es parcialmente completo y correcto.