--------------------TAREA----------------------

DANIELA NICOLÁS LÓPEZ

Ya tenemos la funcion, hacer:

A) $X(n) = X(n-1) + 5$ PARA $n > 1$, $X(1)= 0$

1. n es el argumento de la funcion

2. La operación básica es la multiplicación: $X(n)$

3. No hay operaciones

4. $X(1) = 0$

   $X(n)= X(n-1) + 5$ para $n>1$

5. $X(n) = X(n-1)+5$ sustituir $X(n-1) = X(n-2)+1$

   $= [X(n-2)+5]+5$

   $= X(n-2)+10$ sustituir $X(n-2) = X(n-3)+5$

   $= [X(n-3)+5]+10$

   $= X(n-3)+15$

   $X(n) = X(n-i) + 5*i$ //CASO BASE

Condición incial $X(1)=0$

   $i=n+1$

   5 =constante = c

$X(n)= X(n-i)+c*i$

   $= X(n-n+1)+n$

   $= X(1)+n$

   $= 0+n$

   $=n$

<< ALGORITMO DE ORDEN LINEAL >>

B) $X(n) = 3X(n-1)$ PARA $n > 1$, $X(1)= 4$

1. n es el argumento de la funcion
2. La operación básica es la multiplicación: $X(n)$
3. No hay operaciones
4. $X(1) = 4$

   $X(n)= 3X(n-1)$ para $n>1$

5. $X(n) = 3X(n-1)$ sustituir $X(n-1) = 3X(n-2)$

   $= 3[3X(n-2)]$

   $= 9X(n-2)$ sustituir $X(n-2) = 3X(n-3)$

   $= 9[3X(n-3)]$

   $= 27[X(n-3)]$

   $X(n) = c^i X(n-i)$ //caso base

Condición incial $X(1) = 4$

$i=n+1$

$X(n)= 3^i X(n-i)$

$= 3^n+1 X(n-n+1)$

$= 3^n+1 X(1)$

$= 3^n+1$

>>> ALGORITMO DE ORDEN EXPONENCIAL

C) X(n) = X(n-1) + n PARA n > 0, X(0)= 0

1. n es el argumento de la funcion
2. La operación básica es la multiplicación: X(n)
3. No hay operaciones
4. X(0) = 0

   X(n)= X(n-1)+n para n>0

5. X(n) = X(n-1)+n sustituir X(n-1) = X(n-2)+n

   = [X(n-2)+n]+n

   = X(n-2)+2n sustituir X(n-2) = X(n-3)+n

   = [X(n-3)+n]+2n

   = X(n-3)+3n

   X(n) = X(n-i)+i*n #caso base

Condición incial X(0) = 0

i=n

X(n)= X(n-i)+i*n

= X(n-n)+n*n

= X(0)+n^2

= n^2

>>> X(n) es de orden O(n^2)

D) X(n) = X(n/2) + n PARA n > 1, X(1)=1 (resolver para n=2^K)

1. n es el argumento de la funcion
2. La operación básica es la multiplicación: X(n/2)
3. No hay operaciones
4. X(1) = 1

   X(n)= X(n/2)+n para n>1

5. X(n) = X(n/2)+n   sustituir   X(n/2) = X(n/4)+n

   = [X(n/4)+n]+n

   = X(n/4)+2n   sustituir   X(n/4) = X(n/8)+n

   = [X(n/8)+n]+2n

   = X(n/8)+3n

---------CASO BASE-----------

X(n) = X(n/2^i)+i*n

Condición incial X(1) = 1

Si i = log_2 n -1

=X(n/2^log_2 n) + log_2 n

=X(1)+log_2 n

= log_2 n

6. La eficiencia es de orden log_2 n

//como ejemplo

x(n) = x(n/2/2)+ n/2 + n

```
#este algoritmo no es recursivo
algoritmo misterioso(n):
    s<-0   //t1
    Para i<-1 hasta n hacer    //t2
        s<-s+i*i            //t3
    Fin Para
    devolver s    //t4
Fin
```

i=1	S=0	N=5
1	1	0
2	5	1
3	14	2
4	30	3
5	55	4
6	91	5

s(0)=0

s(n) = t1 + n*t2+ n*t3 + t4

T(n) = (t2+t3)*n + (t1+t4)

T(n) = c*n +c2

Si T(n) = c*n

Algoritmo de orden lineal

¿Qué calcula el algoritmo?

Calcula algo parecido a la sucesión de fibonacci

¿Cuál es la operación básica?

s+n*2+1

¿Cuántas veces se ejecuta la operación básica?

n veces

¿Cuál es la eficiencia del algoritmo?

n

Sugerir una mejora al algoritmo.

Le pondía incrementos