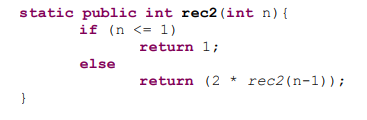
**-Ejercicio 9**

a. Exprese la función del tiempo de ejecución de cada uno de los siguientes algoritmos, resuélvela y calcule el orden.

b. Compare el tiempo de ejecución del método ‘rec2’ con el del método ‘rec1’.

c. Implemente un algoritmo más eficiente que el del método ‘rec3’. (es decir, que el T(n) sea menor).



Expresamos…

cte n<= 1

T(n-1) + cte n>1

n = 8

1° T(n-1) 2° T(n-2) k-esimo T(n - k)

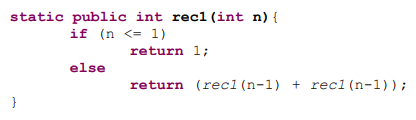
Igualamos en el caso base

n-k = 2 (cuando n sea 2 va a ser la última vez que entra a la recursión)

n - 2 = k

Obtenemos la función entera

T (n) = n - (n - 2) → O (n)



Expresamos…

n <= 1 cte

n > 1 2T(n-1)

1° 2T(n-1)

2° 2 [2T(n-2)] = 4T(n-2)

3° 4(2T(n-2)) = 8T(n-3)

K-esimo 2k T(n-k)

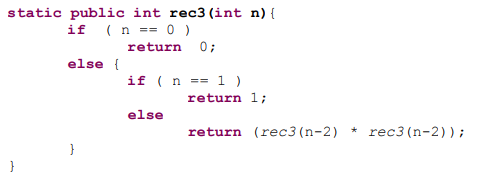
Igualamos en el caso base

n-k = 2 (cuando n sea 2 va a ser la última vez que entra a la recursión)

n - 2 = k

Obtenemos la funcion

T(n) = 2n-2 T(n-(n-2)) → O(2n)



Expresamos…

n = 0 cte

n = 1 cte

n > 1 2T(n - 2)

Paso 1: 2T(n-2)

Paso 2: 2(2T(n-4)) = 4T(n-4)

Paso 3: 2(4T(n-6)) = 8T(n-6)

Paso k: 2k(n - 2k)

Igualamos en el caso base:

n - 2k = 2

n - 2 = 2k

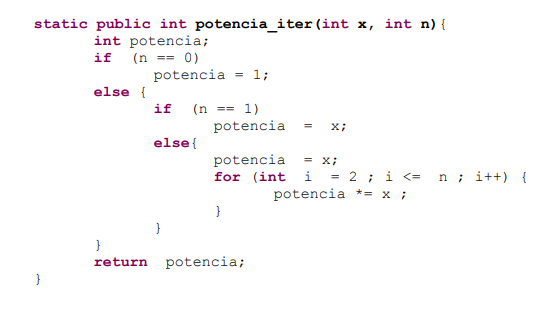
(n-2)/2 = k

n/2 - 2/2 = k

(n/2) -1 = k

Obtenemos la función

T(n) = 2(n/2)-1(n - 2((n/2) -1)) —> O(2n)

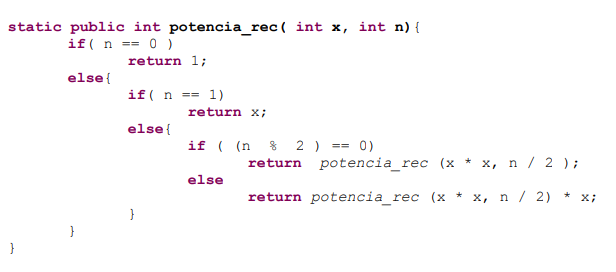


n = 0 cte

n = 1 cte

n > 1 cte +cte → cte + -)cte → cte +(n - 1)cte

→ o(n)



n = 0 cte

n = 1 cte

n % 2 = 0 T(n/2) caso 1

n % 0 != 0 Tn(n/2) caso 2

1° T(n/2)

2° T(n/4)

3° T(n/8)

k- esimo T(n/ 2k)

Igualamos en el caso base

caso 1:

n/ 2k = 0

n = 0

T(n) = (0/2k)

caso 2:

n/ 2k = 1

n = 2k

log2 (n) = k

T(n) = (n/2log(n)) → O(log2(n))

☹️