EJERCICIO 11

BUSCARLINEAL

$T(n) = c_1 + \sum_{i=1}^{n} c_2 = c_1 + n.c_2 \therefore O(n)$

BUSCAR DICOTÓMICA

$$T(n) = \begin{cases} c_1 & , n \leq 1 \\ T(n/2) + c_2 & , n > 1 \end{cases}$$

Paso 1: $T(n) = T(n/2) + c_2$

Paso 2: $T(n) = T(n/4) + c_2 + c_2$

$T(n) = T(n/4) + 2c_2$

Paso 3: $T(n) = T(n/8) + c_2 + 2c_2$

$T(n) = T(n/8) + 3c_2$

Paso i: $T(n) = T(n/2^i) + i.c_2$

$n/2^i = 1$

$n = 2^i$

$\log_2(n) = i$

Reemplazo i: $T(n) = T(n/2^{\log_2(n)}) + \log_2(n).c_2$

$T(n) = T(1) + \log_2(n).c_2 = c_1 + \log_2(n).c_2 \therefore O(\log_2 n)$

Los cálculos del orden de los algoritmos coinciden con la cantidad de iteraciones del punto 2.

EJERCICIO 12

PROCESAR MOVIMIENTOS

$T(n) = c_1 + \sum_{i=1}^{n} (c_2 + \sum_{j=0}^{n} c_3) = c_1 + \sum_{i=1}^{n} (c_2 + n.c_3) = c_1 + n.c_2 + n^2.c_3 \therefore O(n^2)$

PROCESAR MOVIMIENTOS OPTIMIZADO:

$T(n) = c_1 + \sum_{i=1}^{n} c_2 + \sum_{i=1}^{n} c_3 = c_1 + nc_2 + nc_3 \therefore O(n)$

Es claro que el segundo algoritmo va a arrojar resultados más eficientes que el primero al ser de $O(n)$, y el primero de $O(n^2)$.