

1) Función algoritmo  $S(n)$

$cont \leftarrow 2^n$

For  $j \leftarrow 1$  to  $n$  do

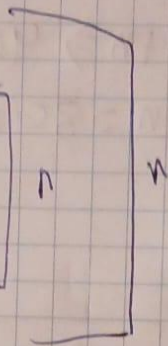
$S \leftarrow cont$   
while  $S > 1$  do

$S \leftarrow S/2$

end while

End for

return  $S$



while

$S = 2^n$

$$S_1 = \frac{S_0}{2} = \frac{2^n}{2^1}$$

$$S_2 = \frac{S_1}{2} = \frac{2^n}{2^{2+1}}$$

$$S_k = \frac{S_{k-1}}{2^k} = \frac{2^n}{2^k}$$

$$\frac{2^n}{2^k} = 1 \Rightarrow 2^n = 2^k \Rightarrow n = k$$

$$T(n) = n \cdot n = n^2$$

$$O(n^2)$$

$$② \quad T(n) = 2T(n/2) + n; \quad T(1) = 1$$

$$① \quad T(n) = 2T(n/2) + n$$

$$② \quad T(n) = 2(2T(n/4) + n/2) + n \Rightarrow 4T(n/4) + 2n$$

$$③ \quad T(n) = 4(2T(n/8) + n/4) + 2n = 8T(n/8) + 3n$$

$$T(n) \Rightarrow \boxed{2^k T(n/2^k) + kn}$$

$$n/2^k = 1 \Rightarrow \boxed{n = 2^k} \Rightarrow \boxed{k = \log_2 n}$$

$$k = \log_2(2^k) \Rightarrow \boxed{\log_2 n = \log_2(2^k)}$$

$$\therefore T(n) = 2^{\log_2 n} \cdot T(1) + n \log_2 n$$

$$= n \cdot 1 + n \log_2 n$$

$$= n + n \cdot \log_2 n$$

$$\Rightarrow O(n) = O(n \cdot \log(n))$$