

Recurrencia:

$$T(n) = 2T\left(\frac{n}{2}\right) + O(n)$$

$$a = 2, b = 2, d = 1, f(n) = O(n)$$

$$a = b^d, = 2 = 2^1 \rightarrow \text{caso 2}$$

$$\text{complejidad final: } T(n) = O(n \log n)$$

Esto significa que el algoritmo es muy eficiente y escalable, que el tiempo de ejecución aumentará si la base de datos crece.