BINARY SEARCH

$$B(n) = \begin{cases} \alpha, & n = \{0, 1\} \\ B(n/2) + C, & n > 1 \end{cases}$$

$i = 1$ $\quad B(n) = B(n/2) + C$

$i = 2$ $\quad B(n) = (B(n/4) + C) + C$

$$B(n) = B(n/4) + 2C$$

$i = 3$ $\quad B(n) = (B(n/8) + C) + 2C$

$$B(n) = B(n/8) + 3C$$

$i = k$ $\quad B(n) = B(n/2^k) + kC$

$$n/2^k = 1$$

$$n = 2^k$$

$$\log n = k$$

$k = \log n$ $\quad B(n) = B(1) + \log_n C$

$$B(n) = \alpha + C \log n \in O(\log n)$$

- IN A BINARY SEARCH, WE DIVIDE THE PROBLEM IN HALF EACH TIME