Time complexity Analysis of Binary Search

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

Now

$T\left(\frac{n}{2}\right) = T\,\frac{\left(n/2\right)}{2} + c$=

$T\left(\frac{n}{2}\right) = T\left(\frac{n}{4}\right) + c$= $\quad T\left(\frac{n}{2^2}\right) + c$

Therefore

$$T(n) = \left[T\left(\frac{n}{2^2}\right) + c\right] + c$$

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

Value of $T\left(\frac{n}{2^2}\right)$ is

$T\left(\frac{n}{2^2}\right) = T\left(\frac{n/2}{2^2}\right) + c$= $T\left(\frac{n}{2^3}\right) + c$

Therefore

$$T(n) = \left[T\left(\frac{n}{2^3}\right) + c\right] + 2c$$

$$T(n) = \ T\left(\frac{n}{2^3}\right) + 3c$$

Let us take $n = 2^k$ therefore $\boldsymbol{k = \log n}$

$$T(n) = T\left(\frac{n}{2^k}\right) + k.c$$

$$T(n) = T\left(\frac{n}{n}\right) + k.c$$

$$T(n) = a. + k.c$$

$$T(n) = \log(n)$$