Ternary Search

↳ Sorted array

↳ Search space into three parts

0	1	2	3	4	5	6	7	8	9
1	2	3	4	5	6	7	8	9	10

1st mid1 2nd mid2 3rd

$mid1 = l + (r-l)//3 = 0 + 9//3 = 3$ $l = 0,\ r = 9$

$mid2 = r - (r-l)//3 = 9 - 9//3 = 6$

$x = 5 \longrightarrow$ searching element

arr(mid1) == x

↳ return mid1

arr(mid2] == x

↳ return mid2

$x = 2$

x < arr(mid 1] → 4

$r = mid1 - 1 \implies$ 1st search space

$x = 9$

x > arr(mid 2] → 7

$l = mid2 + 1 \implies$ 3rd search space

else:

mid1+1, mid2 -1

⟶ 2nd search space

Recurrence Relation (Ternary Search)

$$T(n) = T(n/3) + c$$

$$T(n) = \begin{cases} 1 & n = 1 \\ T(n/3) + c & n > 1 \end{cases}$$

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

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

1

k times

$$n/3^k = 1$$

$$\frac{n}{3^k} = 1$$

$$n = 3^k$$

$$k = \log_3 n$$

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

$$= T\left(\frac{n}{3^{\log_3 n}}\right) + c \cdot \log_3 n$$

$$\Rightarrow T\left(\frac{\cancel{n}}{\cancel{n}^{\log_3 \cancel{3}}}\right) + c \cdot \log_3 n$$

$$\Rightarrow O(\log_3 n)$$

Time complexity of Ternary search