Sorted

arr = [2, 4, 6, 9, 11, 12, 14, 20, 36, 48]

s = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

s = 5, ↓mid = 36 = 8, e = 9

size = 10

target = 36

mid < 36 (mid = 11)

mid < 36 (mid = 20)

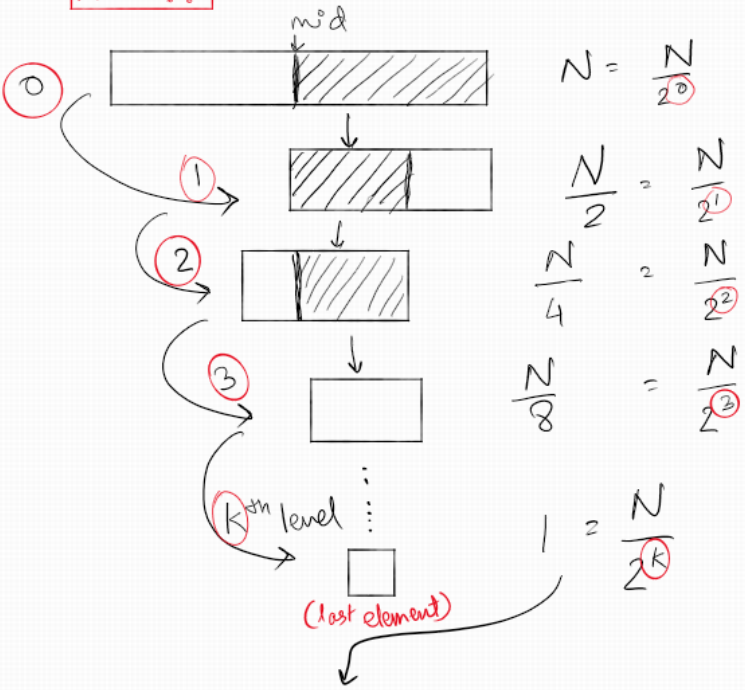
1. Find the middle element ($\frac{s+e}{2}$)
2. if target > mid, then search in the right
   else search in the left
3. if target == mid // ans
4. if s > e // element not found

Time Complexity:

Best Case = O(1) constant

Worst Case = O(log N), N = size of array

HOW??



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

$$\log N = \log(2^k)$$

$$\log N = k \log 2$$

$$K = \frac{\log N}{\log 2} \Rightarrow K = \log_2 N$$

Total comparisons in worst case = log N

Search in 1,000,000 elements:

Linear Search	Binary Search
1,000,000 comparisons	20 comparisons