*Search in Rotated Sorted Array II.

7 7 7 8 8 9 9 9 1 1 2 3 4

l m r

if (t == mid) ⇒ return true.

t = 8 ⇒ search in first half
r = (m-1)

t = 2 ⇒ search in second half
l = m+1

7 8 9 0 1 1 1 2 2 2 2 3 4

l m r

t = 9 ⇒ search in 2nd half
l = mid+1

t = 0 ⇒ search in 1st half
r = mid-1

1 1 1 0 1 1 1 1

both equals to mid

when left element = right element
we can't do predictions to search,
Then we have to move both pointers,
l++, r--

worst case → O(N)
best case → O(log N)

Space → O(1)