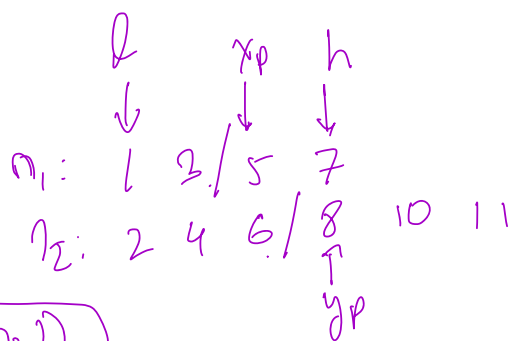


## BINARY SEARCH



5-2

$$x_p = \left( \frac{n_1 + n_2}{2} \right) - x_p$$

if  $l_1 < x_1$  and  $l_2 < x_2$ :  
break

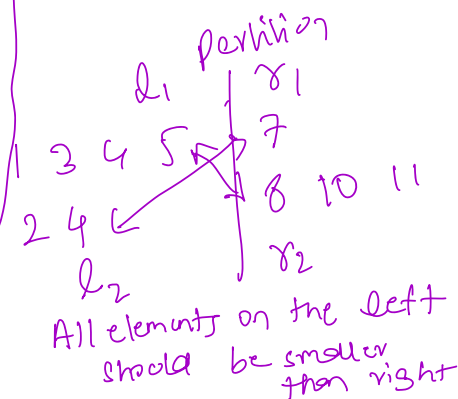
if  $(l_1 > x_2)$ :  
 $h = x_p - 1$

if  $(l_2 > x_1)$ :  
 $l = x_p + 1$

TC-

$$O(\log(\min(n_1, n_2)))$$

Since we should only do binary search on smaller size array from the two.



median:

odd:  $\min(x_1, x_2)$

even:  $\frac{\max(l_1, l_2) + \min(x_1, x_2)}{2}$

NOTE: You don't need to check  $l_1 < x_1$  and  $l_2 < x_2$  since it will always be true in case of sorted arrays.

if  $l_1 > x_2$ , we certainly want to reduce  $l_1$  and/or increase  $x_2$ , since one is complementary of other, they both happen together which means, we will never pick higher value of  $l_1$  to get to final state, so we can prune the partition going right, making complexity  $O(\log n)$ . Similarly, for  $l_2 > x_1$ , we prune left partitions.