BINARY SEARCH

5-2

l $\quad x_p \quad$ h

n_1: 1 3 / 5 7

n_2: 2 4 6 / 8 10 11

y_p

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

if $l_1 < r_1$ and $l_2 < r_2$: break

if $(l_1 > r_2)$:
$h = x_p - 1$

if $(l_2 > r_1)$
$l = x_p + 1$

TC -
$O(\log(\min(n_1, n_2)))$
Since we should only do binary search on smaller size array from the two.

partition

l_1 | r_1

1 3 4 5 | 7

2 4 | 8 10 11

l_2 | r_2

All elements on the left should be smaller than right

median:

odd: $\min(r_1, r_2)$

even: $\dfrac{\max(l_1, l_2) + \min(r_1, r_2)}{2}$

NOTE: You dont need to check $l_1 < r_1$ and $l_2 < r_2$ since it will always be true in case of sorted arrays.

if $l_1 > r_2$, we certainly want to reduce l_1 and/or increase r_2, since one is complimentary 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 $l_2 > r_1$, we prune left partitions.