KTH ELEMENT OF 2 SORTED ARRAYS

☆ In this problem, we are given 2 sorted arrays of integers and our job is to find and return the Kth element of their merged array

Bruteforce solution is to obviously merge both the arrays using a 3rd array and then returning the element at Kth position.

Better solution is to either merge without using any extra space OR to just stop the 2 pointer approach merge as soon as we reach K.

Optimal solution involves Binary Search just like median of two Sorted Arrays. Here, instead of us splitting in two equal halves, we will split in halves of K and N-K.

Pseudocode :

```
kthElement(arr1, n1, arr2, n2, K) {
    if(n1 > n2) {
        return medianSortedArrays(arr2, n2,
                                  arr1, n1, K)
    }
```

```
  low , high = max(0, K - n2), min(k, n1)
  while(low <= high) {
      mid =(low + high) / 2
      oMid = K - mid
      L1 , L2 = INT_MIN, INT_MIN
      r1 , r2 = INT_MAX, INT_MAX
      if(mid  < n1)  r1 = arr1 [mid)
      if (oMid < n2)  r2 = arr2 [oMid]
      if (mid - 1 > 0)  l1 = arr1 [mid-1]
      if (oMid - 1 > 0)  l2 = arr2 [oMid -1]
      if (l1 <= r2  &&  l2 <= r1) {
          if (n1 + n2 % 2 != 0) {
              return  max(l1, l2)
          } else {
              return (max(l1, l2) + min(r1, r2))
                                                /2
          }
      } else if(l1 > r2) {
          high = mid - 1
      } else {
          low = mid + 1
      }
  }
  return 0
}
```