K^{TH} MISSING POSITIVE

☆ In this problem, we are supposed to find and return the K^{th} missing positive integer.

Eg : arr[] = [2 3 4 7 11]
k = 5

Missing numbers are 1, 5, 6, 8, 9 ...
Hence answer is 9

Brute force solution is to run a pass through the entire array. At each step we check if our current element is less than K, if it is we add 1 to K, as it is the displacement by which K moved w.r.t an array of all positive integers. As soon as current element goes beyond K, we return it.

Optimal solution is to somehow find the two elements in b/w our answer will lie. We use binary search to figure this out. At each index, we have a specific number of numbers missing. We just find the indices between which K lies. Like in above example, we know K will lie between index 3 and 4, as 3 numbers are missing at index 3

and 6 at 4. Then we can just go K-3 places at 3.

Pseudocode :

```
kthMissing(arr, N, K) {
    low = 0
    high = N - 1
    while (low <= high) {
        mid = (low + high) / 2
        missing = arr[mid] - (mid + 1)
        if (missing < K) {
            low = mid + 1
        } else {
            high = mid - 1
        }
    }
    return low + K
}
```