

1539 第K个缺失的正整数

Label: 数组

给你一个 严格升序排列 的正整数数组 `arr` 和一个整数 `k` 。

请你找到这个数组里第 `k` 个缺失的正整数。

输入: `arr = [2,3,4,7,11]`, `k = 5`

输出: 9

解释: 缺失的正整数包括 `[1,5,6,8,9,10,12,13,...]` 。第 5 个缺失的正整数为 9 。

输入: `arr = [1,2,3,4]`, `k = 2`

输出: 6

解释: 缺失的正整数包括 `[5,6,7,...]` 。第 2 个缺失的正整数为 6 。

- 遍历

```
class Solution {
    public int findKthPositive(int[] arr, int k) {
        if (k < arr[0]) return k;

        k = k - arr[0] + 1;
        for (int i = 1; i < arr.length; i++) {
            if (k - (arr[i] - arr[i-1]) + 1 > 0) // 如果k还大于0,那就继续遍历
                k = k - (arr[i] - arr[i-1]) + 1;
            else
                return k + arr[i-1]; // 如果 k 小于0 就直接返回
        }
        return arr[arr.length-1] + k;
    }
}
```

- 集合求交集

```
class Solution {
    public int findKthPositive(int[] arr, int k) {
        if (k < arr[0]) return k;

        Set set = new HashSet<>();
        int i = 1;
        while (i <= 1000) set.add(i++);

        for (int j : arr) set.remove(j);

        List<Integer> list = new ArrayList<>(set);
        Collections.sort(list);

        return k-1 >= list.size() ? k + arr.length : list.get(k-1);
    }
}
```

- 二分查找 TODO

```
class Solution {
    public int findKthPositive(int[] arr, int k) {
        if (arr[0] > k) {
            return k;
        }

        int l = 0, r = arr.length;
        while (l < r) {
            int mid = (l + r) >> 1;
            int x = mid < arr.length ? arr[mid] : Integer.MAX_VALUE;
            if (x - mid - 1 >= k) {
                r = mid;
            } else {
                l = mid + 1;
            }
        }

        return k - (arr[l - 1] - (l - 1) - 1) + arr[l - 1];
    }
}
```