# 题目

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。

示例 1：

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 。

示例 2：

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 。

提示：

1 <= arr.length <= 1000

1 <= arr[i] <= 1000

1 <= k <= 1000

对于所有 1 <= i < j <= arr.length 的 i 和 j 满足 arr[i] < arr[j]

# 分析

## 方法一：暴力法

**思路：**

首先最容易想到的莫过于暴力解法，由于1 <= arr[i] <= 1000,1 <= k <= 1000,因而返回的答案不超过2000，不妨把数组开大一些，然后扫描arr，将所有这些出现的元素标记为-1,最后查找第K个不等于-1的元素即可。

**代码：**

class Solution {

public int findKthPositive(int[] arr, int k) {

int i,ret = 0;

int[] ans = new int[2010];

for( i = 1;i <= 2000; i++) ans[i] = i;

for( i = 0;i < arr.length;i++){

ans[arr[i]] = -1;

}

for(i = 1;i <= 2000; i++){

if(ans[i]==-1){

continue;

}else{

k--;

if(k==0)break;

}

}

return ans[i];

}

}

## 方法二：利用arr[i]与其下标i关系

**思路：**

不难发现，一个不缺失元素的序列，会有arr[i]=i+1这样的关系，而在缺失元素之后，会有arr[i]>i+1，简单移项可得arr[i]-i-1 > 0，缺失一个的时候，相差1，两个则相差2，以此类推，缺失越多，两者差距越大，我们要找第k个缺失的，换言之，只要arr[i]-i-1 == k，我们便找到了题目要找的数字。

代码：

class Solution {

public:

int findKthPositive(vector<int>& arr, int k) {

int i,n = arr.size();

for(i=0;i<n;i++){

if(arr[i]-i-1>=k){

return k+i;

}

}

return k+i;//亦可写成：k+n，只不过写成k+i方便理解下面一个解法

}

};

## 方法三：二分查找

**思路：**

然而上述的解法没有用上题目给出的条件严格升序排列，已经找出了 arr[i]-i-1 > 0关系之后，我们可以利用上述的线性查找的方式改为二分查找的方式。

**代码：**

class Solution {

public:

int findKthPositive(vector<int>& arr, int k) {

int left = 0, right = arr.size(), mid = 0;

while(left<right){

mid = left + (right-left)/2;

if(arr[mid]-mid >= k+1){

right = mid;

}else{

left = mid + 1;

}

}

return k + left;

}

};