查找和最小的 K 对数字

给定两个以升序排列的整数数组 nums1 和 nums2 , 以及一个整数 k 。

定义一对值 (u,v)，其中第一个元素来自 nums1，第二个元素来自 nums2 。

请找到和最小的 k 个数对 (u_1,v_1)， (u_2,v_2) … (u_k,v_k) 。

示例 1:

输入: nums1 = [1,7,11], nums2 = [2,4,6], k = 3

输出: [1,2],[1,4],[1,6]

解释: 返回序列中的前 3 对数：

[1,2],[1,4],[1,6],[7,2],[7,4],[11,2],[7,6],[11,4],[11,6]

示例 2:

输入: nums1 = [1,1,2], nums2 = [1,2,3], k = 2

输出: [1,1],[1,1]

解释: 返回序列中的前 2 对数：

[1,1],[1,1],[1,2],[2,1],[1,2],[2,2],[1,3],[1,3],[2,3]

示例 3:

输入: nums1 = [1,2], nums2 = [3], k = 3

输出: [1,3],[2,3]

解释: 也可能序列中所有的数对都被返回:[1,3],[2,3]

```python
class Solution:
    def kSmallestPairs(self, nums1: List[int], nums2: List[int], k: int) ->
 List[List[int]]:
        len_a , len_b = len(nums1), len(nums2)
        dic = {}
        #该索引是为了记录是该和是两个数组中的哪两个元素的坐标
        index = 0
        for i in nums1:
            for j in nums2:
                dic[index] = i + j
```

```
                index += 1
        #将字典按照 value 值（即两个元素和）进行升序
        dic = dict(sorted(dic.items(), key=lambda item: item[1]))
        result = []
        #按照排序结果找到前 k 个组 以及对应元素
        if k > len_a * len_b:
            k = len_a * len_b
        for key in dic.keys():
            #u 在 nums1 中的位置
            i = key // len_b
            #v 在 nums2 中的位置
            j = key % len_b
            result.append([nums1[i], nums2[j]])
            k -= 1
            if k == 0:
                break
        return result
```