

给定两个以升序排列的整形数组 **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**。

找到和最小的 **k** 对数字 **(u1,v1), (u2,v2) ... (uk,vk)**。

示例 1:

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

返回: `[1,2],[1,4],[1,6]`

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:

给出: `nums1 = [1,1,2], nums2 = [1,2,3], k = 2`

返回: `[1,1],[1,1]`

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:

给出: `nums1 = [1,2], nums2 = [3], k = 3`

返回: `[1,3],[2,3]`

也可能序列中所有的数对都被返回:

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

思路一:

brute force的解法, 这种方法我们从0循环到数组的个数和k之间的较小值, 这样做的好处是如果k远小于数组个数时, 我们不需要计算所有的数字对, 而是最多计算k*k个数字对, 然后将其都保存在res里, 这时候我们给res排序, 用我们自定义的比较器, 就是和的比较, 然后把比k多出的数字对删掉即可。

by <https://blog.csdn.net/aishangyutian12/article/details/52013787>

```
class Solution {
public:
    vector<pair<int, int>> kSmallestPairs(vector<int>& nums1,
vector<int>& nums2, int k) {
        vector<pair<int, int>> res;
        for (int i=0; i<min((int)nums1.size(), k); ++i) {
            for (int j=0; j<min((int)nums2.size(), k); ++j) {
```

```
        res.push_back({nums1[i], nums2[j]}); //将所有组合放入返回集
中
    }
}
    sort(res.begin(), res.end(), [](pair<int, int> &a, pair<int,
int> &b) //将所有可能的对按和值大小排序
{    //自定义比较器
    return a.first + a.second < b.first + b.second;
});
if (res.size() > k) //将排序过后的集合中k对之后的对删掉
    res.erase(res.begin()+k, res.end());
return res;
}
};
```