373. Find K Pairs with Smallest Sums

题目描述：https://leetcode.com/problems/find-k-pairs-with-smallest-sums/

> 给定两个按照从小到大排列的数组，求从第一个和第二个随意取数字按照和大小排序，求第k小的结果。
> 例如：
>
> ```
> Example 1:
> Given nums1 = [1,7,11], nums2 = [2,4,6],  k = 3
> Return: [1,2],[1,4],[1,6]
> The first 3 pairs are returned from the sequence:
> [1,2],[1,4],[1,6],[7,2],[7,4],[11,2],[7,6],[11,4],[11,6]
>
> Example 2:
> Given nums1 = [1,1,2], nums2 = [1,2,3],  k = 2
> Return: [1,1],[1,1]
> The first 2 pairs are returned from the sequence:
> [1,1],[1,1],[1,2],[2,1],[1,2],[2,2],[1,3],[1,3],[2,3]
>
> Example 3:
> Given nums1 = [1,2], nums2 = [3],  k = 3
> Return: [1,3],[2,3]
> All possible pairs are returned from the sequence:
> [1,3],[2,3]
> ```

解题思路：

和372类似，维护一个小顶堆，存储当前情况下的最小的结果。
pop出来的结果极为最小（i,j）的，然后再push进去当前的(i+1,j)，其实按照正常想法应该再push(i,j+1),但是这样会和(i-1,j+1)的push结果混掉，所以只push(i+1,j),这样子(i,j+1)就会在(i-1,j+1)时push进去，不影响最终结果。

代码：

```
class Solution {
public:
    vector<pair<int, int>> kSmallestPairs(vector<int>& nums1, vector<int>& nums2,
int k) {
        auto comp = [&nums1, &nums2](pair<int, int> a, pair<int, int> b) {
            return nums1[a.first] + nums2[a.second] > nums1[b.first] + nums2[b.sec
ond];};
        vector<pair<int, int> > res;
        if(nums1.size() == 0 || nums2.size() == 0 || k == 0) {
            return res;
        }
        priority_queue<pair<int, int>, vector<pair<int, int> >,  decltype(comp)> q
(comp);
        q.push(make_pair(0, 0));
        int t = k;
        while(t-- > 0 && !q.empty()) {
            int i = q.top().first;
            int j = q.top().second;
            q.pop();
            res.push_back(make_pair(nums1[i], nums2[j]));
            if(i != nums1.size() - 1)
                q.push(make_pair(i+1,j));
            if(i == 0 && j != nums2.size() - 1)
                q.push(make_pair(i,j+1));
        }
        return res;
    }
};
```