```cpp
class Solution {
public:
    vector<vector<int>> kClosest(vector<vector<int>>& points, int k) {
        priority_queue<pair<double, vector<int>>> maxHeap;

        for (const auto& point : points) {
            double distance = sqrt(point[0] * point[0] + point[1] * point[1]);

            maxHeap.push({distance, point});
            if (maxHeap.size() > k) {
                maxHeap.pop();
            }
        }

        vector<vector<int>> result;
        while (!maxHeap.empty()) {
            result.push_back(maxHeap.top().second);
            maxHeap.pop();
        }

        sort(result.begin(), result.end(),
             [](const vector<int>& a, const vector<int>& b) {
                 double distA = sqrt(a[0] * a[0] + a[1] * a[1]);
                 double distB = sqrt(b[0] * b[0] + b[1] * b[1]);
                 return distA < distB;
             });

        return result;
    }
};
```