#define REP(i, n) for(int i = 0; i < n; i++)

#define REPP(i, n) for(int i = 1; i <= n; i++)

#define ALL(obj) (obj).begin(), (obj).end()

class Solution {

public:

vector<vector<int>> outerTrees(vector<vector<int>>& points) {

sort(ALL(points));

int n = points.size();

if(n < 3) return points;

vector<vector<int>> u, l;

u.push\_back(points[0]); u.push\_back(points[1]);

for(int i = 2; i < n; i++){

while(u.size() >= 2){

int m = u.size();

if(cross(u[m - 1], u[m - 2], points[i]) > 0) u.pop\_back();

else break;

}

u.push\_back(points[i]);

}

l.push\_back(points[n - 1]); l.push\_back(points[n - 2]);

for(int i = n - 3; i >= 0; i--){

while(l.size() >= 2){

int m = l.size();

if(cross(l[m - 1], l[m - 2], points[i]) > 0) l.pop\_back();

else break;

}

l.push\_back(points[i]);

}

map<vector<int>, int> mp;

reverse(ALL(l));

for(vector<int>& v: l) mp[v]++;

for(int i = u.size() - 2; i >= 1; i--) if(mp[u[i]] == 0) l.push\_back(u[i]);

return l;

}

int cross(vector<int> p, vector<int> q, vector<int> r){

int ax = r[0] - p[0];

int ay = r[1] - p[1];

int bx = q[0] - p[0];

int by = q[1] - p[1];

int res = ax \* by - ay \* bx;

return res;

}

};