#include <iostream>

#include <algorithm>

using namespace std;

int heap1[100001], heap2[100001];

void Swap(int a, int b){

int temp = heap1[a];

heap1[a] = heap1[b];

heap1[b] = temp;

}

void heap2\_Swap(int a, int b){

int temp = heap2[a];

heap2[a] = heap2[b];

heap2[b] = temp;

}

void max\_heap1(int root, int bottom){

int max\_child;

int left\_child = root \* 2 + 1;

int right\_child = root \* 2 + 2;

if(left\_child <= bottom){

if(left\_child == bottom)

max\_child = left\_child;

else {

if(heap1[left\_child] >= heap1[right\_child])

max\_child = left\_child;

else

max\_child = right\_child;

}

if(heap1[max\_child] > heap1[root]){

Swap(max\_child, root);

heap2\_Swap(max\_child, root);

max\_heap1(max\_child, bottom);

}

}

}

void heap1\_sort(int size){

int index;

for(index = size / 2 - 1; index >= 0; index--){

max\_heap1(index, size - 1);

}

for(index = size - 1; index > 0; index--){

Swap(index, 0);

heap2\_Swap(index, 0);

max\_heap1(0, index - 1);

}

}

int main(void) {

int n, count = 1;

cin >> n;

for(int i=0; i<n; i++){

cin >> heap1[i];

cin >> heap2[i];

}

heap1\_sort(n);

for(int i=0; i<n; i++){

if(heap1[i] == heap1[i+1] && (i != n-1)){

count++;

}

else if (count > 1){

sort(heap2 + i + 1 - count, heap2 + i + 1);

count = 1;

}

}

for(int i=0; i<n; i++){

cout << heap1[i] << " ";

cout << heap2[i] << "\n";

}

return 0;

}