Laboratory work #3

Student: CAO Xinyang  
Student ID: 20321308  
Timus Name: hduads2022\_20321308

Mail: c.x\_yang@foxmail.com

Problem #1207

Screenshot from Timus:



Explanation of algorithm:

By comparing the coordinate values, the point at the bottom left corner is chosen as the origin. Find the angles of the other points to the origin (by the atan2 function) and sort these angles(Quicksort is used in the code). The (N / 2)th point is the other point we need. The final answer is the point we found and the origin.

Computational complexity of algorithm:

F(N) = N + N + N + NlogN = 3N + NlogN

T(N) = O(NlogN)

Source code:

import java.util.Scanner;

public class App {

public static void quickSort(double[] arr, int[] index, int l, int r) {

int i, j, temp1, t1;

double temp, t;

if (l > r) {

return;

}

i = l;

j = r;

temp = arr[l];

temp1 = index[l];

while (i < j) {

while (temp <= arr[j] && i < j) {

j--;

}

while (temp >= arr[i] && i < j) {

i++;

}

if (i < j) {

t = arr[j];

arr[j] = arr[i];

arr[i] = t;

t1 = index[j];

index[j] = index[i];

index[i] = t1;

}

}

arr[l] = arr[i];

arr[i] = temp;

index[l] = index[i];

index[i] = temp1;

quickSort(arr, index, l, j - 1);

quickSort(arr, index, j + 1, r);

}

public static void main(String[] args) throws Exception {

Scanner scan = new Scanner(System.in);

int N = scan.nextInt();

int[] X = new int[N];

int[] Y = new int[N];

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

X[i] = scan.nextInt();

Y[i] = scan.nextInt();

}

int origin = 0;

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

if (X[i] < X[origin] || X[i] == X[origin] && Y[i] < Y[origin]) {

origin = i;

}

}

double[] atg = new double[N - 1];

int[] index = new int[N - 1];

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

atg[i] = Math.atan2((X[i] - X[origin]), (Y[i] - Y[origin]));

index[i] = i;

}

for (int i = origin + 1; i < N; i++) {

atg[i - 1] = Math.atan2((X[i] - X[origin]), (Y[i] - Y[origin]));

index[i - 1] = i;

}

quickSort(atg, index, 0, N - 2);

int a = origin + 1;

int b = index[N / 2 - 1] + 1;

String ans = a + " " + b;

System.out.println(ans);

scan.close();

}

}