Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Algorithms Lab HS20
Department of Computer Science
Prof. Dr. E. Welzl
cadmo.ethz.ch/education/lectures/HS20/algolab

Threefold Problem Sheet – Partitions

This sheet presents three problems which share several characteristics and as a result they appear quite similar in nature. But this does not necessarily mean that these problems can be solved using the same strategies and techniques. In fact, the whole idea of these threefold sheets is to highlight how possibly subtle differences and changes in a problem formulation can have a great impact on how the resulting problem can be solved. We believe that studying these sheets is an excellent practice of how to approach the modeling aspects of solving a problem.

Read through the problems and try to work out by yourself possible lines of modeling and then algorithmically solving them. Give reasons of why or why not certain approaches appear plausible. Ideally, substantiate your ideas by sketching corresponding proof ideas.

This sheet was handed out and discussed during the tutorial on December 2, 2020.

Introduction (for all problems)

You are given a set P of n points in the plane. Some pairs of the points (we call them *edges*) are connected by straight line segments. The *length* of an edge is the Euclidean distance of the two points. Moreover, you are given two distinguished points $q, r \in P$, $q \neq r$. A q-r *partition* of the set P is a pair of sets $(S, P \setminus S)$ such that $S \subset P, q \in S, r \in P \setminus S$. An edge $\{u, v\}$ is *crossing* the partition $(S, P \setminus S)$ iff $u \in S$ and $v \in P \setminus S$.

Exercise – *MiniSum*

The goal of this exercise is to find a q-r partition $(S, P \setminus S)$ of the points P which minimizes the **sum** of **squared lengths** of all edges crossing the partition. The sum of squared lengths of all edges crossing a partition is defined to be zero if no edge is crossing the partition.

In this exercise, there is an edge between every pair of points whose distance is at most d.

Input The first line contains $1 \leqslant t \leqslant 30$, the number of testcases. Each of the t testcases is described as follows:

- It starts with a line that contains two space-separated integers n, d ($2 \leqslant n \leqslant 500$, $0 \leqslant d \leqslant 2^{25}$).
- Each of the next n lines contains two space-separated integers x_i, y_i ($|x_i|, |y_i| \leqslant 2^{24}$), specifying the coordinates of the i-th point $(x_i, y_i) \in P$. The points q and r are the first and second point in the input.

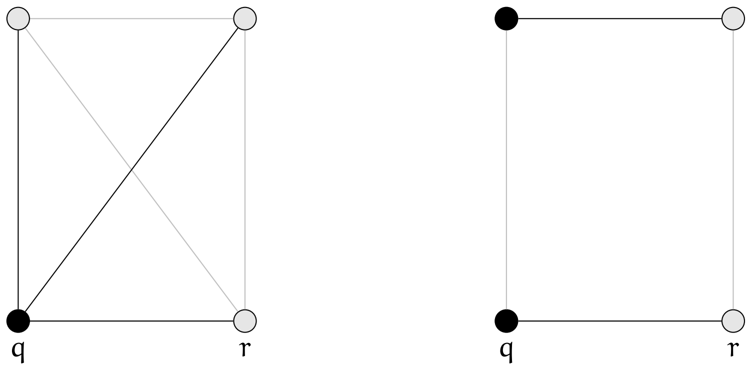
Output For every test case output a single line that contains an integer specifying the number of points in the set S, followed by the points in the set S represented by indices (1-based) according to the order in which the points appear in the input. If there are multiple solutions, you can output any of them.

Sample Input

```
2
4 5
0 0
3 0
3 4
0 4
4 4
0 0
3 0
3 4
0 4
```

Sample Output

```
1 1
2 1 4
```



(a) First test case.

(b) Second test case.

Figure 1: Sample test cases. An optimal partition and edges crossing it are marked by shading.

Exercise – *MiniLength*

The goal of this exercise is to find a q-r partition $(S, P \setminus S)$ of the points P which minimizes the **maximum length** of an edge crossing the partition.

In this exercise, there is an edge between every pair of points.

Input The first line contains $1 \leqslant t \leqslant 30$, the number of testcases. Each of the t testcases is described as follows:

- It starts with a line that contains a single integer n ($2 \leqslant n \leqslant 10^3$).
- Each of the next n lines contains two space-separated integers x_i, y_i ($|x_i|, |y_i| \leqslant 2^{24}$), specifying the coordinates of the i-th point $(x_i, y_i) \in P$. The points q and r are the first and second point in the input.

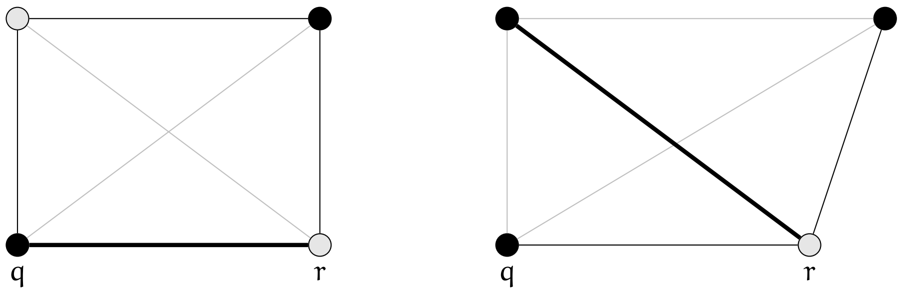
Output For every test case output a single line that contains an integer specifying the number of points in the set S, followed by the points in the set S represented by indices (1-based) according to the order in which the points appear in the input. If there are multiple solutions, you can output any of them.

Sample Input

```
2
4
0 0
4 0
4 3
0 3
4
0 0
4 0
5 3
0 3
```

Sample Output

```
2 1 3
3 1 3 4
```



(a) First test case. (b) Second test case.

Figure 2: Sample test cases. An optimal partition and edges crossing it are marked by shading. The longest edge crossing the partition is highlighted in bold.

Exercise – *MiniDist*

Remark. This problem is more difficult and may need additional ideas.

The goal of this exercise is to find a line ℓ separating[1] the points q and r which minimizes the **maximum Euclidean distance** of a point $p \in P$ to the line ℓ.

Input The first line contains $1 \leqslant t \leqslant 30$, the number of testcases. Each of the t testcases is described as follows:

- It starts with a line that contains a single integer n ($2 \leqslant n \leqslant 10^4$).
- Each of the next n lines contains two space-separated integers x_i, y_i ($|x_i|, |y_i| \leqslant 2^{24}$), specifying the coordinates of the i-th point $(x_i, y_i) \in P$. The points q and r are the first and second point in the input.

[1] i.e., either at least one of q and r is on ℓ or they are on different sides of ℓ

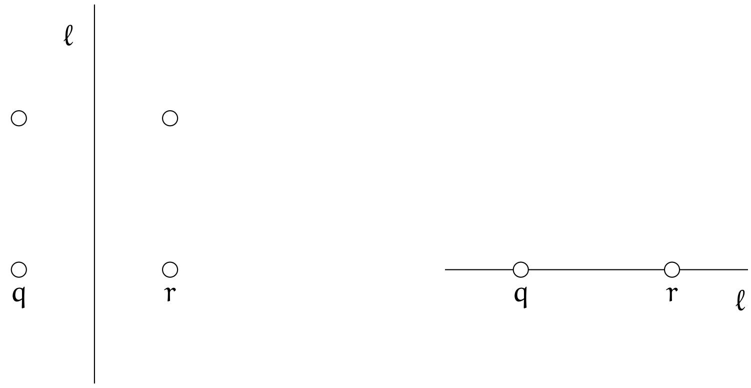
Output For every test case output a single line containing three space-separated integers a, b, c specifying the parameters of an optimal line $\ell : ax + by + c = 0$. If there are multiple solutions, you can output any of them.

Sample Input

```
2
4
0 0
2 0
2 2
0 2
2
0 0
2 0
```

Sample Output

```
-1 0 1
0 1 0
```



(a) First test case. (b) Second test case.

Figure 3: Sample test cases.