Problem A. Beautiful Partitioning Challenge

Input file name: `standard input`
Output file name: `standard output`
Time limit: from 10 s to 30 s
Memory limit: 1024 MB

Formal statement

You are given a roadmap data as a planar graph G. Each node of the graph is given with its geographical coordinates (latitude and longitude). It is guaranteed that no pair of given nodes is on a distance more than 100 kilometers from each other while all nodes have pairwise distinct coordinates.

Each road is given as a sequence of node IDs. It is a polyline on a sphere without self-intersections or self-loops. It is guaranteed that if two road polylines share a common point on a sphere, this point is necessary one of the given nodes of the roadmap.

The edges of graph G are formed by union of edges (segments between consecutive nodes) of all roads.

The given roadmap graph G is not guaranteed to be connected.

We call a geographical point *internal* point of G if:

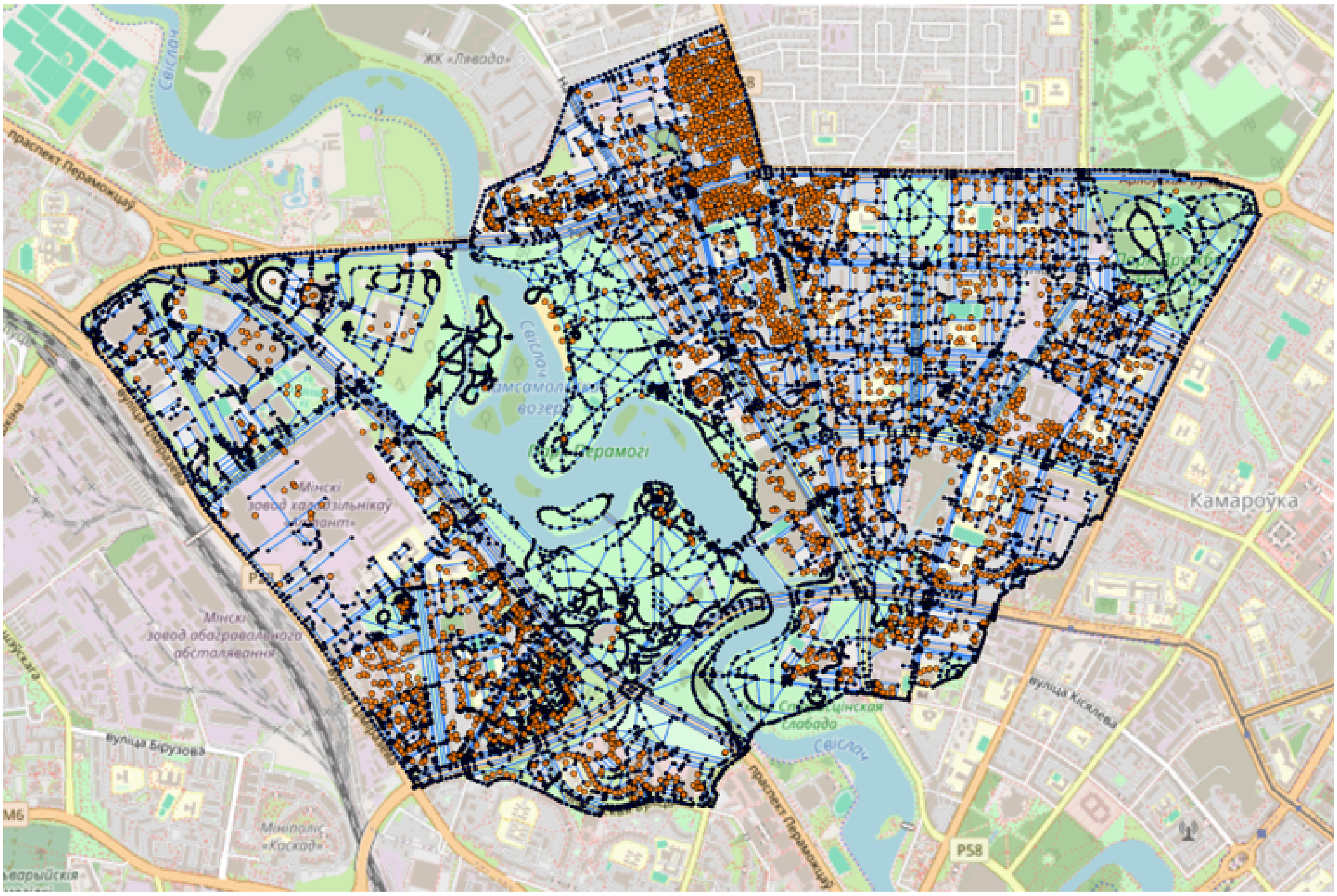
- It does not coincide with any node of the roadmap;
- It does not belong to any road of the roadmap;
- It does not belong to the outer side of the roadmap planar graph, that is, it belongs to an area enclosed by a finite set of roads.

Additionally, you are given a set of points called *users*, each one described by a its geographical coordinates. For every user it is guaranteed that it is an internal point of G.

You should find k sequences of node IDs of G satisfying the following:

- **Each sequence represents a simple cycle of G. This implies that it represents a polygon on a 3D-sphere, or a spherical polygon.**
- **Every internal point of G should be an internal point of exactly one of k polygons.**
- **Every polygon should contain at least 512 and at most 1024 users.**

Note that k is not given in the input, but is part of your output and can vary. The score of your solution does not explicitly depend on the value of k.



Graphical representation of a possible input over a real-world map. Roadmap planar graph is represented in blue (dark blue points represent nodes and light blue segments represent edges) and users are represented by orange large points.

Input

The first line contains the number of nodes n $(1000 \le n \le 700000)$ of graph G.

The following n lines contain the node id $(0 \le id \le 1000000)$ and its geographical coordinates lat and lon.

The next line contains the number of edges m $(3 \le m \le 700000)$ of graph G.

The following m lines contain two integers u_i, v_i $(0 \le u_i, v_i \le 1000000, u_i \ne v_i)$, denoting the endpoints of i-th edge of the input graph.

The next line contains the number of $users$ t $(512 \le t \le 700000)$.

The following t lines contain three values u_i, lat_i, lon_i $(1 \le u_i \le t)$, denoting the id and geographical coordinates of i-th user.

In the input data, the maximum number of digits after the floating point does not exceed 15.

Output

In the first line of output, print k - number of sequences of node IDs of G. Then for the i-th sequence, print the number of nodes l_i that it contains. Then print the i-th sequence of node IDs of G. Then, print the number of users that contains i-th polygon. Then, print the i-th sequence of user IDs.

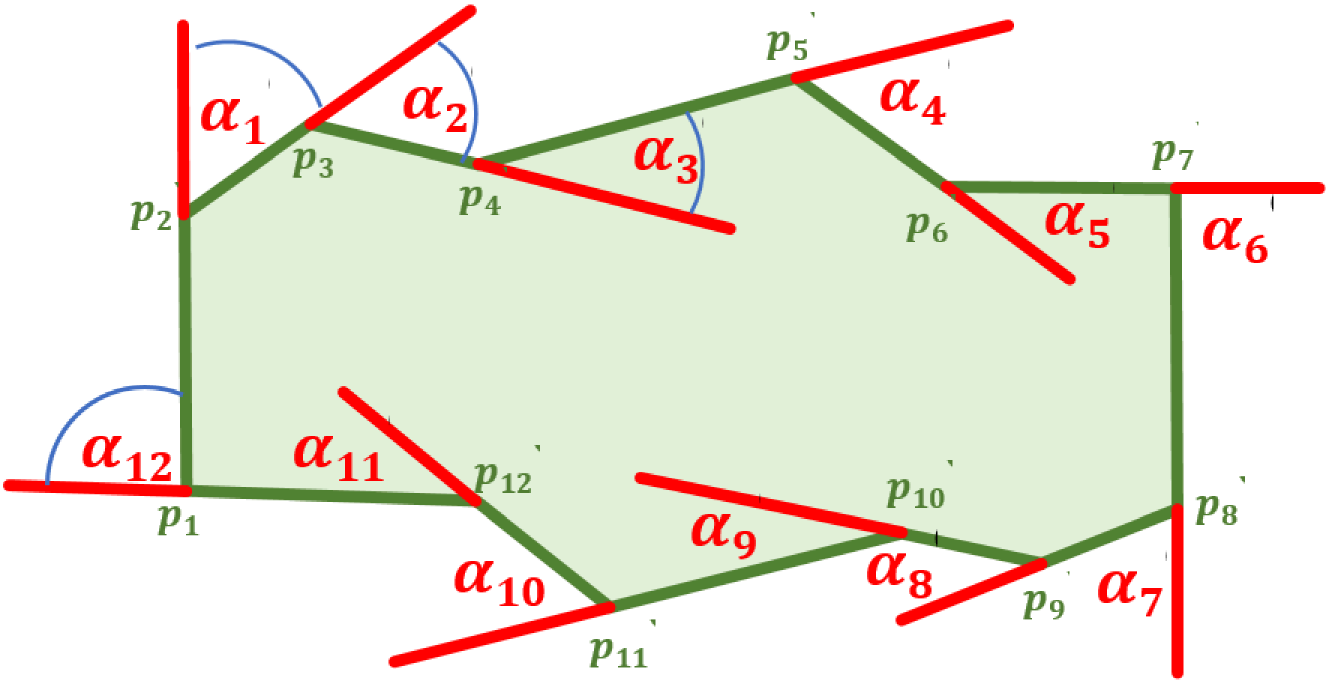
Scoring

Score of your output is determined by the *greatest Absolute Curvature (AC)* of polygons in your output. The Absolute Curvature of a polygon P consisting of points $p_1, p_2, \ldots, p_n$ (in transversal order) in 3-dimensional space (converted from longitude-latitude pairs to points on the 3D-sphere) is defined as (with $p_{n+1} = p_1$ and $p_{n+2} = p_2$):

$$AC(P) = \frac{1}{2\pi} \sum_{i=1}^{n} \alpha(\overline{p_i p_{i+1}}, \overline{p_{i+1} p_{i+2}})$$

where α is the absolute value of the shortest angle between two vectors that can be computed with the formula (arccosine of the value of dot product divided by the product of vectors' lengths):

$$\alpha(\overline{A}, \overline{B}) = \arccos \frac{\overline{A} \cdot \overline{B}}{\|\overline{A}\| \|\overline{B}\|}$$



Absolute Curvature evaluation for a sample polygon P consisting of 12 points. The resulting $AC(P)$ equals to the sum of all 12 angle values divided by 2π.

Then, the total score of the solution is given by

$$Score(P_1, P_2, \ldots P_k) = \max_i AC(P_i)$$