Contest: Round 1

Normal view

Problem: CutTheRoots

Problem Statement

Several seeds were planted in one jar. The grown-up plants have to be separated and moved to other jars, one plant per jar. While growing, the plants developed a massive intertwining root system which has to be separated as well with as little damage as possible.

The jar with plants is represented as a circle on a plane. Each plant is represented as a 2D root system, since we are not interested in its stem (it doesn't need to be cut when the plants are separated).

The root system of one plant starts with a plant base point - the point where root system connects to the stem. Each root grows either from the base point of a plant or from the end of another root.

To separate the plants, you need to cut the soil in the jar. Each cut is modeled as an infinite straight line which goes through points (X1, Y1) and (X2, Y2). Cut lines cut every root they intersect. Two plants are considered to be separated as long as there exists at least one line which separates their base points.

Your code must implement a single method int[] makeCuts(int NP, int[] points, int[] roots). The parameters give you the following information:

- **NP** - the number of plants.
- **points** - the coordinates of endpoints of roots. Endpoint i (0-based) has coordinates (point[2*i], point[2*i+1]). Note that there will be more than **NP** points in this array; the first **NP** points will be base points of plants.
- **roots** - the roots, represented with indices of their endpoints. Root j (0-based) connects endpoints roots[2*j] and roots[2*j+1]. It is guaranteed that each root endpoint belongs to exactly one plant (i.e. no chain of roots exists which would connect base points of two different plants). Roots of one plant or different plants can intersect each other, but not share an endpoint.

You must return a int[] representing a set of at most 4*NP lines which separate all plants. If your return describes L lines, it must contain exactly 4*L elements, line k (0-based) is defined as going through points (ret[4*k], ret[4*k+1]) and (ret[4*k+2], ret[4*k+3]).

The score for an individual test case is the total length of roots which stay attached to the base point of their plants, divided by total length of all roots before cutting. If there exist two plants which are not separated with the cuts, or the return value was invalid in any other way, the score for that test case is 0. Your overall score will be the sum of YOUR/MAX over all test cases, where YOUR is your raw score on a test case, and MAX is the maximal score achieved by anyone on that test case (test cases on which you scored 0 don't contribute to the sum).

Tools and Information

An offline tester is available here. You can use it to test/debug your solution locally. You can also check its source code for exact implementation of test case generation and score calculation. That page also contains links to useful information and sample solutions in several languages.

Definition

Class:	CutTheRoots
Method:	makeCuts
Parameters:	int, int[], int[]
Returns:	int[]
Method signature:	int[] makeCuts(int NP, int[] points, int[] roots)

(be sure your method is public)

Notes

- This match is rated.
- The time limit is 10 seconds and the memory limit is 1024MB for a single test case.
- The number of plants (NP) will be less than 105 and there will be at least 5 plants. There will be less than 105000 roots and at least 50.
- X1,Y1,X2 and Y2 must be between 0 and 1024. All plant and root coordinates will also fall within this range.

Examples

0)

```
seed = 1
NP = 5
```

```
NR = 500
Length of roots = 5280.447005971856
```

1)

```
seed = 2
NP = 28
NR = 22400
Length of roots = 135720.29520142596
```

2)

```
seed = 3
NP = 98
NR = 66248
Length of roots = 533797.0490723713
```

3)

```
seed = 4
NP = 10
NR = 8520
Length of roots = 57235.583661379584
```

4)

```
seed = 5
NP = 31
NR = 16647
Length of roots = 121573.53376857964
```

5)

```
seed = 6
NP = 11
NR = 2761
Length of roots = 24576.521680862115
```

6)

```
seed = 7
NP = 39
NR = 21138
Length of roots = 186372.3779337002
```

7)

```
seed = 8
NP = 100
NR = 16400
Length of roots = 192691.78233601913
```

8)

```
seed = 9
NP = 27
NR = 13905
Length of roots = 104341.6522919012
```

9)

```
seed = 10
NP = 100
NR = 60000
Length of roots = 539048.1828607273
```

This problem statement is the exclusive and proprietary property of TopCoder, Inc. Any unauthorized use or reproduction of this information without the prior written consent of TopCoder, Inc. is strictly prohibited. (c)2010, TopCoder, Inc. All rights reserved.