演算法Final Report(第八組)

**組員**：b97502022張浩軒 b00901024張瑞宇 b00901079陳威宇 b00901086林宏驊

**題目**：國際積體電路電腦輔助設計軟體製作競賽題目E Region Query

Our project try to implement the region query problems in the CAD contest.

1. Introduction to Problem Definition

Basically, in this region query problem, several polygons are drawn on a two-dimensional plain. Then, some question about regions on the plain are asked.

Then, some question about regions on the plain are asked:

Find the number of polygons contained by a given (rectangular) window area.

Find the number of polygons interacting with a given window area.

Find the number of polygons interacting with given window that do not overlap with others interacting with the window.

Find the number of polygons whose areas are less than a given number.

Find the percentage of the given window area that is covered by polygons.

Their definitions are specified below:

* 1. Problem 1: Find the number of polygons contained by a given (rectangular) window area.

A certain window (such as the red one in the figure above) is given. The number of polygons that are totally contained inside the windows have to be found (two of them, in the case above).

* 1. Problem 2: Find the number of polygons interacting with a given window area.

Interacting means that two polygons have overlapped region, overlapped edges, or overlapped points. In the above figure, polygons shaded green either overlap with the window area, or have some point or edge touching the window.

The number of polygons interacting with the window area have to be found.

* 1. Problem 3: Find the number of polygons interacting with given window that do not overlap with others interacting with the window.

Overlap here means that the two polygons of question have nonzero overlapping area. Therefore, simply touching, or, overlapping region are only edges or single/multiple points are not considered as overlapping here. Overlap and interacting are two different relation of polygons.

In this problem, we have to find out the number of polygons interacting with the given window that does not overlap with other polygons.

* 1. Problem 4: Find the number of polygons whose areas are less than a given number.

This is a rather straight forward problem, in which we identify the polygons having area smaller than the specified value.

* 1. Problem 5: Find the percentage of the given window area that is covered by polygons.

In the above figure, we have to find out the percentage of the window area that is also covered by polygons (that is, the percentage of yellow area that is also painted blue).

* 1. Restriction of polygons and input format

The polygons are simple polygons, that is, they do not contain holes.

1. Core Algorithm

The major challenge of this region query problem is that polygons may overlap each other, and edges may intersect each other. There is a choice to merge overlapping polygons into larger polygons. Such choice would still require knowledge of intersection of edges, and it raises a problem that the merged polygons may not be simple polygons. That is, they may contain holes, which makes it difficult to represent them easily by vertex lists, and their area cannot be calculated by the same simple method utilized on simple polygons. Thus, to deal with the central part of our problem, we use sweep line algorithm and and Bentley-Ottmann algorithm as our core algorithm.

1. Problem Solving

Characterizing the problems:

Based on their characteristics, we categorize the five problems as follows:

Problem 4, finding the polygons with area less than the specified value, only deals with each polygon itself: its own area. No information among different polygons are required.

The remained problems ask about the relationship between the polygons and a certain given window area. Problem 1 and 2, finding polygons contained inside or interacting with the given window area, filters out the polygons that are closely involved, which helps problem 3 and 5, for the same window area. Therefore, we categorize problem 1 and 2 as utilities, which would be used by problem 3 and 5.

Problem 3 and 5, dealing with overlapping of area and intersection of edges, is our core problem.

To solve problems efficiently, data structure of polygons should also be constructed.

Our problem solving strategy is as follows:

* 1. Preset : construct data structure

Data of each polygons are contained in class Polygons. It contains:

* + 1. Area (used in problem 4):

The polygons area. In problem 4, it would be compared with the specified maximum area value.

* + 1. Vertex list (used in problem 1, 2, and constructing edge list):

List of the x and y coordinates of the vertices. This is directly obtained from the input file, and the ordering of vertices is either clockwise or counterclockwise, depending on the input file.

* + 1. Min enclosing rectangle (used in problem 1)

We record the y coordinate value of the highest and lowest points, and the x coordinate value of the leftmost and rightmost points. These four value determine the horizontal and vertical range of the polygon, and is equivalent to the min enclosing rectangle.

* + 1. Edge lists (used in problem 3, 5)

These are several edge lists, which together contains all the edges of the polygon except for vertical ones. They are constructed from the vertex list, and their usage and explanation will be explained with problem 3 and 5, where they are actually used.

* 1. Problem 4: Find the number of polygons whose areas are less than a given number.
     1. Preset: find the area of each polygon

To find th area of each polygon, take some point (in our implementation, we take the first vertex in our vertex list) as a fixed point A, then 

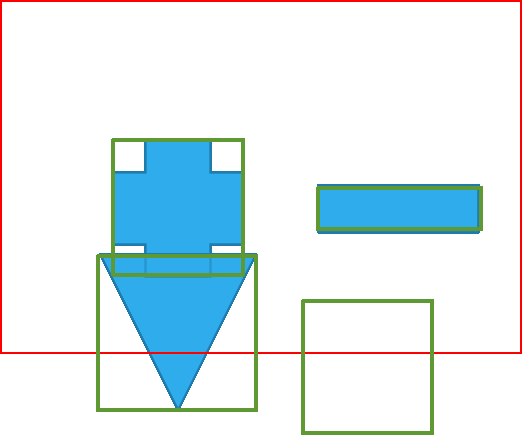
Then, compare the specified value with the area of each polygons.

* + 1. Compare the specified value with the area of each polygons

Computing the area of one polygon.

* 1. Problem 1: Find the number of polygons contained by a given (rectangular) window area.

We examine the polygons’ min-enclosing rectangles. The window containing the min-enclosing rectangle of a polygon is equivalent to the window containing the polygon.



We have to test for each polygon’s min-enclosing rectangle. Therefore, it is a linear search problem, with complexity O(P).

* 1. Problem 2: Find the number of polygons interacting with a given window area.

Firstly, a polygon interacting with a given window area must have its min-enclosing rectangle interacting with the given window area (either overlapping or touching). Therefore, we can eliminate those that have their min-enclosing rectangle not interacting with the given window area.

Then we examine the remaining candidates. The polygons may be contained inside the window, may contain the window, or none of the above.

1. Edge intersection occurs.

If the polygon is of the third type, then some of its edges would intersect with the window edge. Therefore, the second thing we do is to examine each edge of the polygon, and test whether each of them intersect with the window edge. If an intersection occurs, interaction is guaranteed.

1. Polygon contained inside the window

Still the remaining are the cases of containing or contained. The contained cases can be obtained from problem 1.

As for the cases where the window is contained by the polygon, extra examination is required.

1. Window contained inside the polygon

For the window to be contained inside the polygon, firstly, the window must be contained inside the min-enclosing rectangle of the polygon.

For further examination,

