Data structures

The data structure which used in here is binary space partitioning tree(BSP). The reason to use this kind of data structure is, it can divide the data into smaller units until the partitioning satisfies the given two requirements. The python list is used also used in here to store the line segments because it is faster than linked list.

Encountered problems

The first problem that i had whether which is the data structure do i want to use to store the line segments. Solution was a BSP tree. As described above, it is the appropriate data structure to this scenario. The next problem was how should i determine a line is in front of or back of another line. To solve this i used geometric equations. First i calculated the formula of the first line. Then substituted the coordinates of second line to that formula. If the results of both coordinates are positive, then the second line is in front of the first line. If the both lines are negative, then the second line is in the back of the first line. If one result is negative and the other is positive, then the lines are intersecting.

Next problem was how should i identify the lines that face each other. The method i used in here was, whether the lines are in front of other or not. For example assume line1 and line2 are two separate lines. If one line1 is in front of line2 and the line2 is in front of the second line1, then the two lines are face to face.

Building tree data structure

First, I created a node to hold data, reference to left and right children. Then in the tree, there is a root node which holds the whole line system as its data. Then S^{th} is selected as the main line. Rest of the lines are divided into two lists (front and back) with respect to the main line. Added those lists into two separate nodes. Connecte those new nodes as the front child and the back child of root node. Then replace the data of the root node as S^{th} line(main line). Then take the front list and check whether the main line is in the front of all the front list lines. If at least one line is failed the check, then continue the same process

recursively to the front node and to the back node. If main line is in front of the all lines, then the process should be stopped. Because all of those front list lines including main line are facing each other.

Retrieving results

Front most line

Front most line should be situated at the front most child of the tree. To get the line which is in front of all lines, i traverse to the front node of the tree and return that line.

Face to face lines

Face to face lines can be found in the tree like this. Nodes that have more than one child have face to face lines.

See the positions of all lines

I did a inorder traversal to get the positions of all lines back to front.