Training Project I: Corner Stitching

Input: Rectangle list files. Each line describes one rectangle by listing its four corners sequentially.

Operations:

0. Read all rectangles and save them in a list *RctList*. Determine the boundaries of your corner stitching plane based on all input rectangles.

1. Read one rectangle from input rectangle list *RctList* and use area search with current rectangle as search region to verify if any one block tile in the corner stitching plane overlap current rectangle. If yes, skip current rectangle and return to Step 1 to read next rectangle. If no, add current rectangle to corner stitching plane and return to Step 1 to read next rectangle until all rectangles are processed. During Step 1, you have to count the total numbers of rectangles to be inserted into the corner stitching plane and their total area.

2. Use the bounding box of your corner stitching plane as an area and use enumeration operation to enumerate all blocked tiles in the plane. Whenever you visit a blocked tile, count its number and sum up its area. Use the number and the total area to verify if they match the numbers obtained in Step 1.

3. Read one rectangle from input rectangle list *RctList* and remove it from the corner stitching plane. You have to make sure current rectangle is in the plane. Repeat the same operations until all rectangles are removed from the plane. Finally the plane should have only one free tile. If it is hard to debug the bugs of tile removing, you can perform enumeration operation whenever one rectangle is removed from the plane. With this, you can know which removal operation brings troubles to your program.

note: report your runtime and memory for your execution.