**CSE2046 Project 3**

Ahmet TURGUT Alperen KÖKER

150117046 150117035

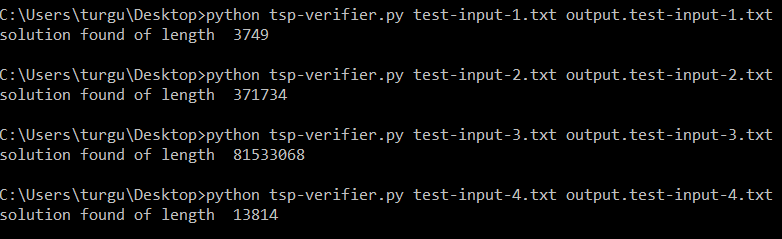
**Features of the algorithm**

Firtsly created the object for the cities and we keep the id and x , y coordinats and visited value.After created the arraylist and fill the arraylist with cities. After filling arraylist Origin found by averaging x and y coordinates.

Our goal is to divide it into eight equal area So tangent value was found for all points and divided into areas. For all this area have a own arraylist. For all arraylist founded a own path and paths are mediatized.

Program found a starter and finish cities for a path. Starter city is the closest city to line of area. Finish city is the closest city to other line of area. Program have a recursive method for a find a neighbor (closest city) city. After the call the method founded city added the path and visited value change a true .Hereby program forbid the circle in the path .And repeatedly call the method and add all the cities. There is only one city left(finish).And program add manually the finish city to path. After program found a path each area , areas are connected with each other's starter and finish points. Hereby program found a big path. I simply schematized the path finding algorithm.

**Output:**



Two features stand out so that we can understand how good the algorithm is.

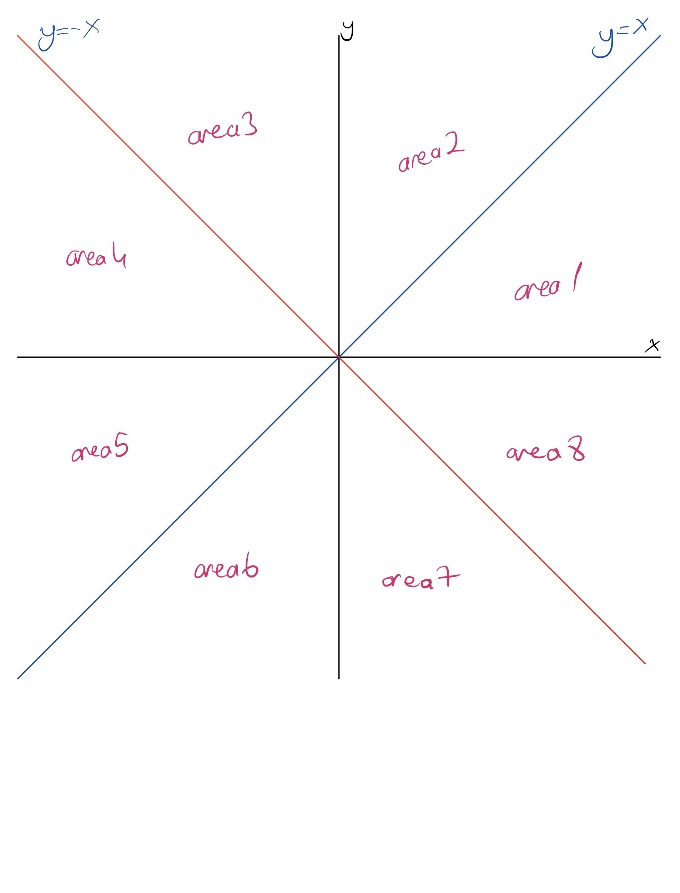
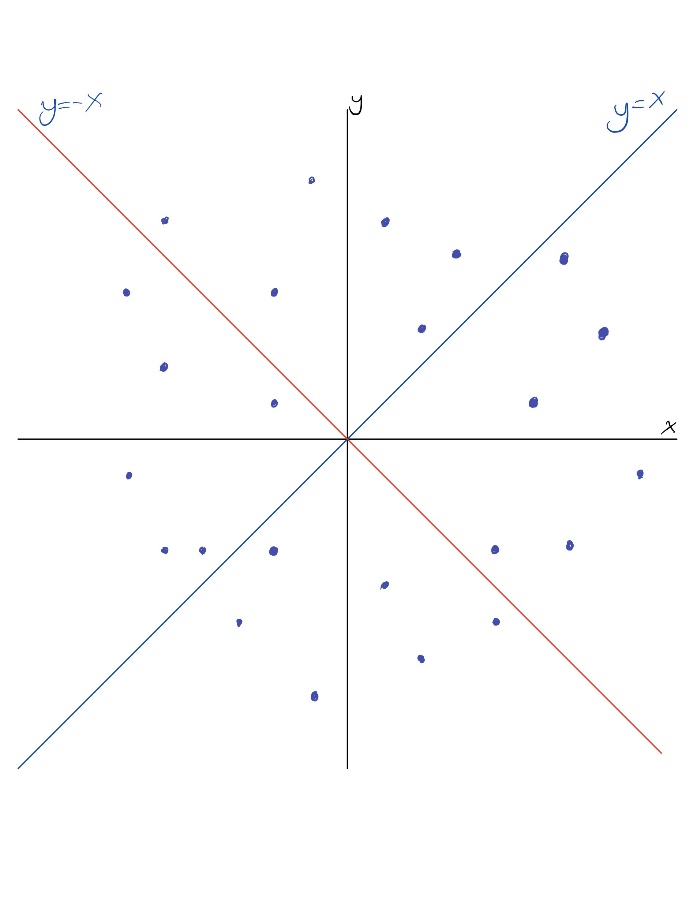
* time efficiency
* space efficiency

In this algorithm, we managed both situations efficiently.

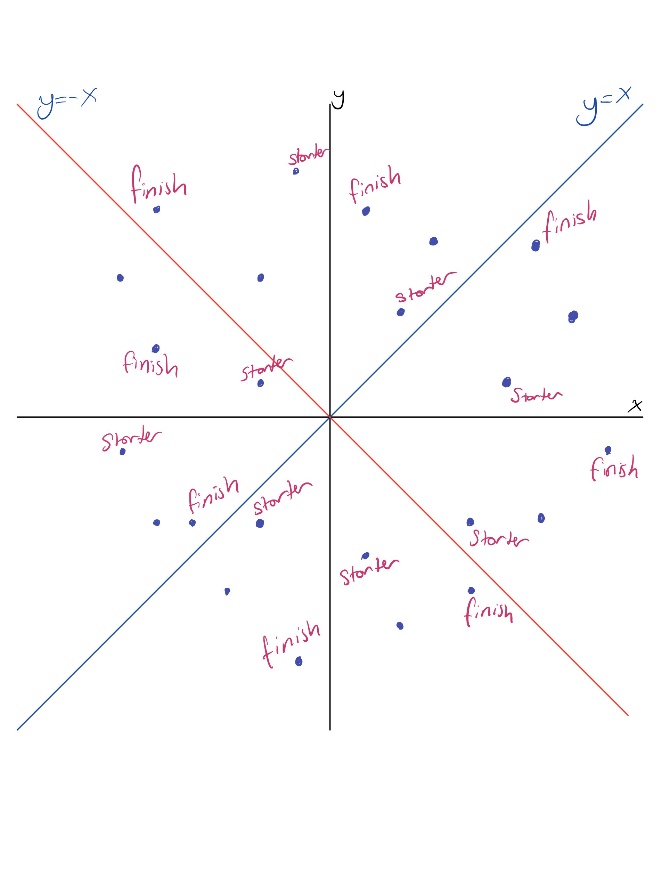
**Division Of Labor:**

We did it by giving ideas together using the online zoom application.

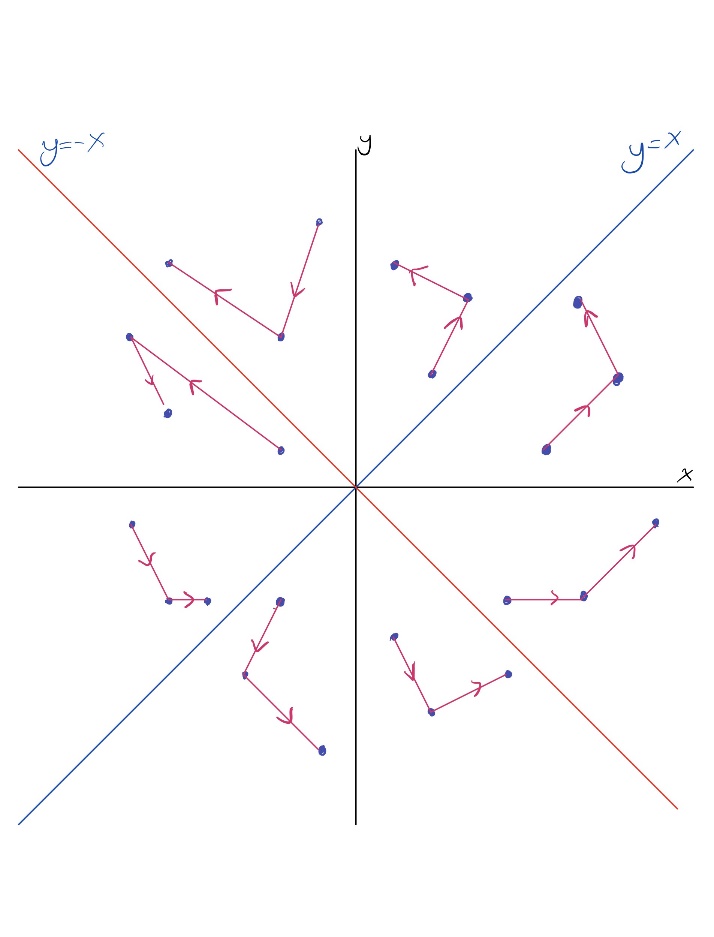
**A schematic representation of how the algorithm works is attached.**

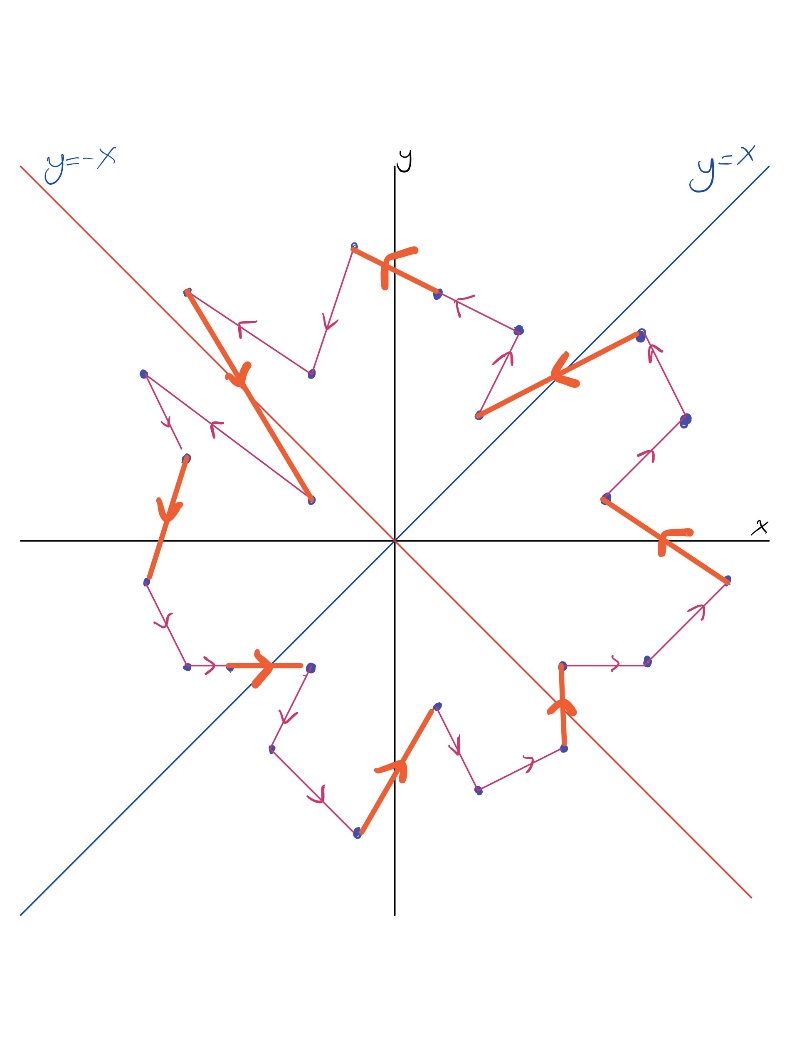
First of all, an origin point was determined according to all coordinates. According to this origin point, the coordinate system was divided into 8 regions. Cities were placed in these regions.



These areas require a starting point and an end point. By accepting these points as start and end, paths were created in these areas. These start and end points were calculated according to the tangent values of the points.



Paths were created in a recursive way by going to the nearest unvisited city.



These paths are combined with the start and end points of the regions. As a result of this merge, a general path was obtained. This general path created a way to visit.