Conception and Implementation of Spatial Analysis Methods

Homework No.7

Raptakis Giorgos
MN: 374030

Contents

Used Functions

For the solution of Divide and conquer- algorithm we created firstly some functions.

Function of:vector<point>random_points(intn, intmin, intmax) which create a random number of points in a given interval.

Function of: double distance(point, point);which find the distance between two points.

Function of:doubleMinDistThreePoin(vector<point>);Which calculates the minimum distance of three points

Function of:doubleStripMinDist(vector<point>);this function accepts as arguments a vector of points which belong in a strip. Sorts the points as regards their y-value, and finds with a for-loop the minimum distance.

Function of: doubleTwoHalf(constvector<point>points)this function accepts as arguments a vector of points and returns the value of the minimum distance.

Divide and conquer- algorithm

Through the function random_pointswe create in our example 20 random points in the interval [0,50], and then we sorted them with increasing in regard to x-value.

Then we call the function of doubleTwoHalf(vector<point>points) As input have a vector of sortedpoints. It is a recursive function which split the vector at the media point and call again itself so as to calculate a minimum distance of the right and of the left vector of points. When the vector size is smaller or equal than three, then call the function of MinDistThreePoin() and calculates the minimum distance. Then is calculated the minimum distance of the left and the right part. A vector of points is created, which belongs in the strip with center the media point and width the minimum distance. Then is calculated the lowest distance of points inside the strip calling the function of StripMinDist(strip). After the comparison of minimum distances the lowest one is returned.

Results

The implementation of the same data with the two different ways, Sweep line algorithm (Lab5) and Divide and conquer algorithm (Lab7) has the same results of the closest pair of points, using the same data. The difference is observed in the proceed time. The Sweep line algorithm takes 2371 units of time and the Divide and conquer algorithm returns value of 105 units of time. So, we observe that the latest method is much more faster.

In our firstexample, we create 20 random points in the interval of 0 to 50, which saved with name final.txt file and visualized in QGIS software.

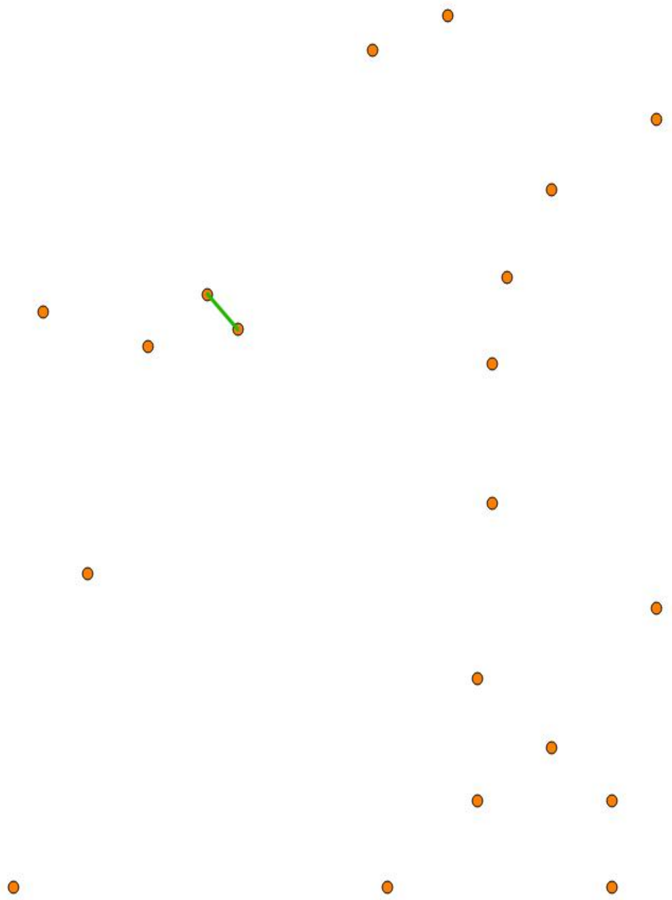

Figure1: Closest pair of points of 20 points