Algorithm and Data structures coursework.

Algorithm one – Nearest Neighbour

General formula.

1. Initialize current point as starting point.
2. While points left in list  
   2.1. Add current point to list.  
   2.2. For each point left in list  
    2.2.1 If distance between current point and ‘for point’ < previous distance  
    2.2.2 Closest point is this point.  
    2.2.3 Previous distance is now distance between current point and ‘for point’  
   2.3 end for
3. Remove closest point from list.
4. Current city is now the closest point.
5. End while.

Pros

* Easy to implement.
* Good for small data sizes.
* Next element will be there. (when looking through all elements).

Cons

* Requires large storage of data.
* Large searching problem (Have to iterate over list until list is empty).
* Assumptions made about distance.
* Brute force.

Possible improvements

* Swap order in couples of two’s? does this provide a better result?

Intro

The travelling salesman problem (known as tsp) is a question which asks “In a set of cities, what is the shortest distance possible that visits each city only once and also returns to the origin?”. The main issue of this question is the that the number of cities is subject to how many you want to travel between.

Initial attempt

* Nearest Neighbour algorithm.
* Mention stuff above.

Improvements

Results

NN not good. – display average time and polygon that mapped route.

NN v1 ??. – display average time and polygon that mapped route (should be the same ^^).

NN v2 ??. - display average time and polygon that mapped route.

NN v3 (v1+v2) . - display average time and polygon that mapped route.

Good/bad about improvements. Why?

Conclusion – I hate reports