Project 3: TSP – Closest Edge Insertion Heuristic

- Learning objectives. At the completion of this project, you should be able to
  - Implement a greedy algorithm for solving TSP

- Background
  - A Traveling Salesperson Problem (TSP) is an NP-complete problem. A salesman is given a list of cities and a cost to travel between each pair of cities (or a list of city locations). The salesman must select a starting city and visit each city exactly one time and return to the starting city. His problem is to find the route (also known as a Hamiltonian Cycle) that will have the lowest cost. (See http://www.tsp.gatech.edu for more info)

- Problem
  - Solve an instance of a TSP problem by using a variant of the greedy heuristic
  - Build tour by starting with one vertex, and inserting vertices one by one.
  - Always insert vertex that is closest to an **edge** already in tour.
  - Data for each problem will be supplied in a .tsp file (a plain text file).

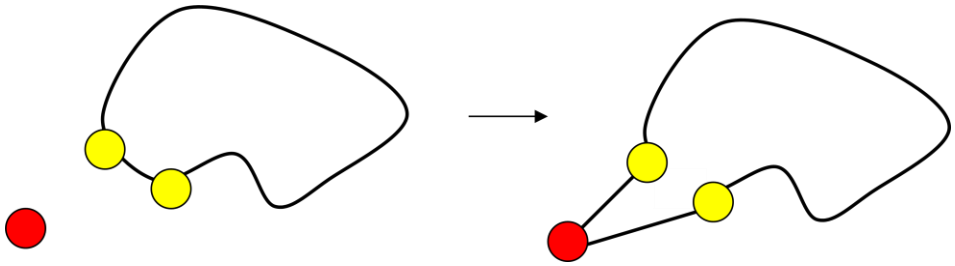

Figure from: www.cs.uu.nl/docs/vakken/an/an-tsp.ppt

- Hints
  - An instance of TSP with < 4 nodes is trivially solvable.

- Deliverables
  - Project report (3-4 pages). Describe how you selected initial nodes and generated the list of edges to consider. How did you selected the "next" node to be added. Show the route and the cost of the best tour for each provided dataset as well as order in which nodes have been added to the tour. How does this algorithm compare to other approaches for solving TSP you have tried so far? How quick is this method?
  - Well-commented source code for your project. You can use any language you like, but I reserve the right to ask you to demo performance of your algorithm on a new dataset.
  - You have to include a GUI with visual representation of the solutions for this project.

- Equation of a line (http://webmath.com/equline1.html)
- Distance from Point to a Line (http://mathworld.wolfram.com/Point-LineDistance2-Dimensional.html)