Dr. Roman V. Yampolskiy
CECS-545

Project 2: TSP – Search with BFS and DFS

- Learning objectives:
  - Search Techniques for graphs
  - BFS and DFS algorithms

- 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.
    For this lab we are looking at a special case of TSP in which not all cities are connected and the salesperson only needs to find the best path to a target city not visit all cities.
- Problem
  - For the given dataset (11PointDFSBFS.tsp), starting at the first city (city 1) find the shortest path to the goal city (city 11).
    - Implement Breadth First Search (BFS) and Depth First Search (DFS) algorithms
  - Visit cities in numerical order if you need to break a tie. You can hardcode connected edges into your algorithm for this problem, see table below

Table 1: Cities connected by a one way path of Euclidian distance (left = from, top = to).

pt	1	2	3	4	5	6	7	8	9	10	11
1		x	x	x							
2			x								
3				x	x						
4					x	x	x				
5							x	x			
6								x			
7									x	x	
8									x	x	x
9											x
10											x

- Deliverables
  - Project report (3-4 pages) describing results of your experiments and your implementation. Which algorithm was faster in finding the target city? How long did it take (time and transitions)?
  - 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 don't have to include a GUI with visual representation of the solutions for this project, but it might be useful for your future TSP related projects in this course.