



## CSE 3353 ALGORITHMS: LAB 3

**Overview:** Using both naive brute force and dynamic programming (DP) techniques, solve the traveling sales man problem (TSP) (find a Hamiltonian circuit) for a given list of nodes and positions (graph).

Code Link: <https://classroom.github.com/a/1ZT9glX7>

**Due: November 04, 2020 @ 11:59pm.**

### **Code Requirements:**

- Always use Node 1 as your start/finish point of TSP
- Read node list in from a text file which uses same format as the positions.txt file from Lab 2
  - NodeID x, y, z  $\rightarrow$  1,1.23,3.45,4.112  $\rightarrow$  int,float,float,float
  - You can derive your own list of nodes and positions
  - Code will be tested against a private list of nodes/positions. Code will only be tested up to the number of nodes identified in writeup as total nodes able to analyze.
- Return optimal (shortest) path for given list of nodes
- Start with a list of 4 nodes and increase to the maximum number that is feasible with your computer for both naïve and DP algorithms
- Design your code to be reusable for new algorithms, incorporating the file loader and output system into their own independent single interfaces. Utilize patterns (or modified patterns) to make informed decisions on how to architect your project. How your code was designed will hold a substantial weight on overall project grade.

### **Report Requirements:**

- Plot timing with excel for each algorithm in a single graph. X axis should be number of nodes in graph, Y axis should be total timing.
  - Also provide a table to summarize data in graph
- Provide a detailed review of results obtained and be sure to explain time complexity for each algorithm
  - Provide a graph that shows a comparison of timing captured on code execution vs the asymptotic timing determined for given algorithm
- Explain and document your design decision. Should include write up that explains architecture decisions and how you designed for functionality and extensibility as well as a UML diagrams.
- Explain/derive subproblems and how the dynamic programming portion was developed. This should be an in depth description or problem breakdown (sub problems, etc.), how algorithm is used, how it satisfies the problem/sub-problems and improves on brute force timing.

***All code used in project should be in the “Code” folder, raw data files generated should be in “Data” and your final report should be in “Report” folder***