CS 581 Spring 2024 Written Assignment #01

Due: **Sunday, March 3, 2024, 11:59 CST**

Points: **100**

Instructions:

1. Use this document template to report your answers. Name the complete document as follows:

   LastName_FirstName_CS581_WA01.doc or pdf

   ONLY PDF or MS Word file formats will be accepted.

2. Submit the final document to Blackboard Assignments section before the due date. No late submissions will be accepted.

Objectives:

1. (100 points) Implement and evaluate two search algorithms.

Input data file:

Your input file will be a CSV (comma-separated values) file (see Programming Assignment #01 folder in Blackboard - campus.csv).

You **CANNOT** modify nor rename input data files.

Each row in that input file will correspond to STATE information: STATE_LABEL, and state 2D Cartesian space coordinates: X and Y to be positive integers.

Input (four states in this case, but there will be more) file (text file) format:

$$A, X_A, Y_A$$
$$B, X_B, Y_B$$
$$C, X_C, Y_C$$
$$D, X_D, Y_D$$

Where:

- A, B, C, D are state LABELS,
- X_i, Y_i are state coordinates.

Your submission should include:

- Python code file(s). Your py file should be named:

  cs581_P01_AXXXXXXXX.py

where AXXXXXXXX is your IIT A number (this is REQUIRED!). If your solution uses multiple files, make sure that the main (the one what will be run to solve the problem) is named that way and others include your IIT A number in their names as well.

- this document with your results and conclusions. You should rename is to:

LastName_FirstName_CS581_Programming01.doc or pdf

Problem description: You are given a weighted complete graph G (example shown on Fig. 1). Your task is to solve a Traveling Salesman Problem [TSP] (find minimum cost/weight Hamiltonian Cycle) on G using algorithms specified below.

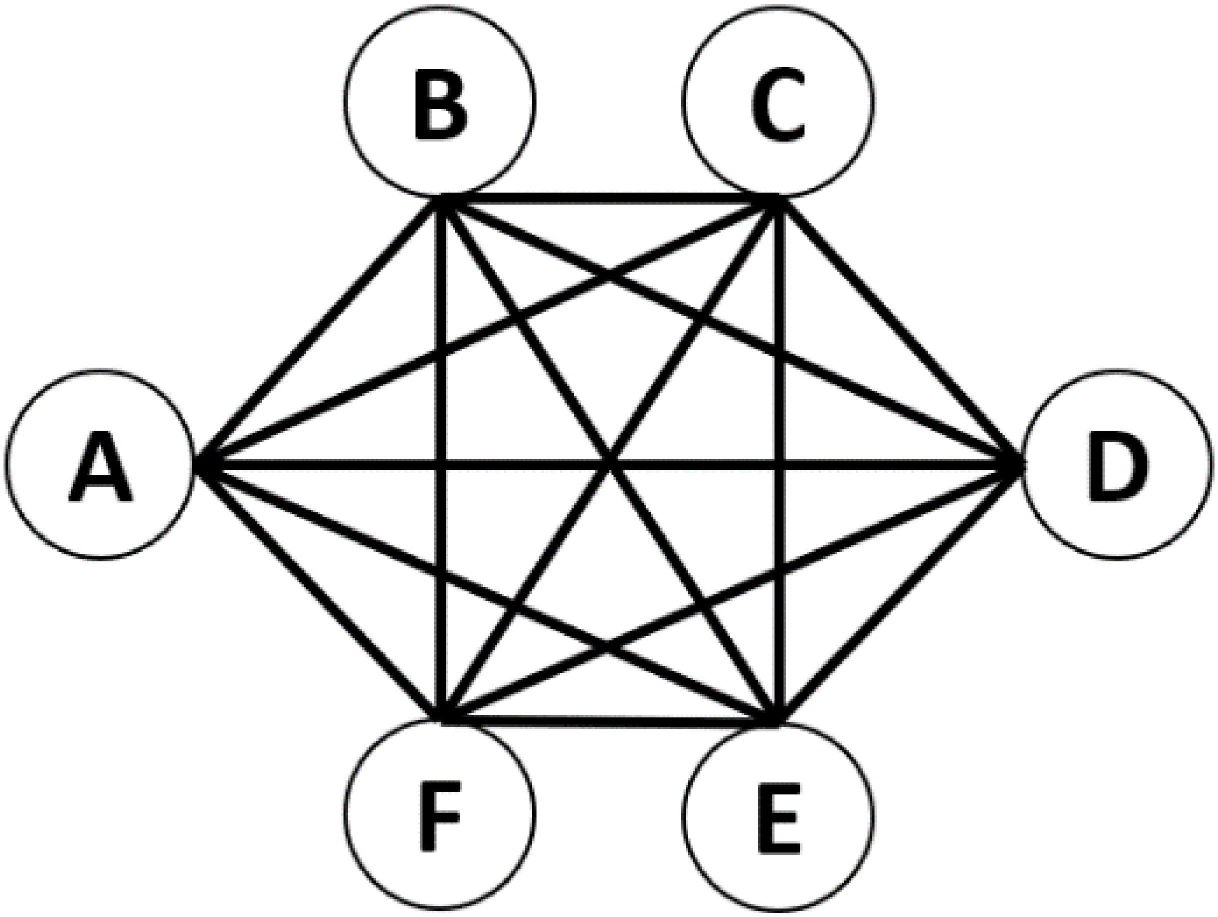

Figure 1: *Sample complete (fully connected, weights NOT shown graph G.)*

Assume that edge weights represent **straight line distances between states.**

Your task is to implement two search algorithms in Python:

- Simulated Annealing, and

- Genetic Algorithm,

and apply them to solve the TSP (**starting at Stuart Building [SB] node**) problem using provided data.
Your program should:

- Accept four (4) command line arguments corresponding to two states / state capitals (initial and goal states) so your code could be executed with

  python cs581_P01_AXXXXXXXX.py FILENAME ALGO P_1 P_2

  where:

  - cs581_P01_AXXXXXXXX.py is your Python code file name,
  - FILENAME is the input CSV file name (graph G data),
  - ALGO is mode in which your program should operate
    - 1 – Simulated Annealing
    - 2 – Genetic Algorithm
  - P_1 is a value for a specific algorithm parameter:

    Simulated Annealing P_1 is the initial temperature T value,

    Genetic Algorithm P_1 is the number of iterations K,
  - P_2 is a value for a specific algorithm parameter:

    Simulated Annealing P_2 is the α parameter for the temperature cooling schedule,

    Genetic Algorithm P_2 is the mutation probability P_m value,

  Example:

  python cs581_P01_AXXXXXXXX.py DATA.csv 2 1000 0.01

  If the number of arguments provided is NOT four, your program should display the following error message:

  ERROR: Not enough or too many input arguments.

  and exit.

- Load and process input data file provided (assume that input data file is ALWAYS in the same folder as your code - this is REQUIRED! DO NOT HARDCODE YOUR LOCAL FILE PATH). Make sure your program is **flexible enough to accommodate different input data set** (with a different [size, nodes, edges, etc.] graph, but structurally the same). Your submission will be tested using a different file!