CSE2046 Assignment 2

Half Traveling Salesman Problem (H-TSP)

Due date: June 6th, 2023

The travelling salesman problem (TSP) asks the following question: Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city? In this assignment, we will focus on a modified version of TSP, Half TSP, where the salesman should visit exactly half of the cities and return back to the origin city. You may start from any city.

Inputs are: n cities, with their locations (x and y coordinates) in a 2D grid.

Output is: Ordering (tour) of half of these cities (($\lceil n/2 \rceil$ cities), such that total distance to travel is minimized.

Distance between two cities is defined as the Euclidian distance rounded to the nearest integer. In other words, you will compute distance between two cities $c_1 = (x_1, y_1)$ and $c_2 = (x_2, y_2)$ as follows:

$$d(c_1, c_2) = round\left(\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}\right)$$

For example, if three cities are given with the coordinates $c_1 = (0,2), c_2 = (2,3)$ and $c_3 = (3,0)$, then a tour with ordering $c_1,\ c_2,\ c_3,\ c_1$ has a total distance of :

$$rnd(\sqrt{5}) + rnd(\sqrt{10}) + rnd(\sqrt{13}) = rnd(2{,}23) + rnd(3{,}16) + rnd(3{,}60) = 2 + 3 + 4 = 9$$

Project Specification:

Your team (3 or 4 students) is asked to design and implement a method for finding a tour which is as close to the optimal tour as possible. Note that TSP problem is an NP-hard problem, and H-TSP is at least as hard as ordinary TSP problem (you also need to decide which cities to select). Therefore it is too hard to find optimal solutions. Your goal is not to design an algorithm for the optimal solution, but you are requested to do your best. This is an open-ended project.

You may do the following:

- Read as much as you want to learn about how to solve the problem. But **you have to cite** any resources you use. You may want to start with some approximation algorithms (such as local search heuristics) in chapter 12.
- You may use whichever programming language you want.

You may **not use** the following:

- Existing implementations or subroutines
- Extensive libraries (if you are not sure, check with the instructor)
- Other people's code.

Input format: Inputs will always be given to you as a text file. Each line defines a city and each line has three numbers seperated by a white space. The first number is the city ID, second number is the city's x-coordinate and third number is the y-coordinate.

Output format: You must output your solution into another text file with $\lceil n/2 \rceil + 1$ lines where n is the number of cities. The first line will include length of the tour you find. The next $\lceil n/2 \rceil$ lines should contain the city identifiers in the order they are visited by your tour. Each city must be visited exactly once in this list. If your output is not in this valid format, you will not receive any credit from your solutions.

Important note: We will provide some sample inputs, and a verifying procedure in Google classroom. This verifying procedure will be verify that your output is a VALID output for a given input. (It will not verify the optimality of the code, but only its validity). We will provide it in few days. Using the verifier, you should test that your program report the correct total distance. Also note that, your code should work in reasonable time for the inputs with up to 50.000 cities.

Test instances: On June 5th, we will announce 4 test instances at the web site. By June 6th, 23:59, you will be required to submit 4 seperate output text files (according to the output format) corresponding to each of these test instances**.**

Project report: Together with the output text files, you should also submit a project report. The project report should describe the ideas behind your algorithm as completely as possible. It should not exceed 3 pages in length in no less than 10pt. **You should also describe the "division of labor – who did what?".**

Grading policy:

60% of your grade will be determined by your project report. Clarity and creativity of your work will significantly affect your grade. (Note: If you just implement a simple nearest neighbor algorithm with $\lceil n/2 \rceil$ cities you can only get half of this grade)

40% of your grade will be determined by your solutions to the test instances. Studies that find better solutions will get higher grades.

Come on to the contest: "Hodri Meydan!"

For each of the four test inputs, we will indentify best, second best and third best performing teams. Best, second and third projects receive 3 stars, 2 stars and 1 star, respectively. So if a Project performs best in all four test inputs it will receive 3*4=12 stars.

Stars	1	2	3	4	5	6	7	8	9	10	11	12
Bonus Points	5	7	9	11	13	15	17	19	21	23	25	30