

312
CS 312: Artificial Intelligence Lab
Assignment 5

Problem Statement

Given a set of cities (coordinates) and distances between them, find your best(shortest) tour (visiting all cities exactly once and returning to the origin city) in a given amount of time, viz. **Travelling Salesman Problem.**

Description

- **Input** - You will be given files in the following format.
 - First line will contain either *euclidean* or *noneuclidean* indicating whether the distances between the cities are Euclidean or not.
 - Second line will contain the number of cities (**N**). E.g. 100 (Indices 0 - 99)
 - Next **N** lines will contain the two-dimensional coordinates (space separated) of the cities.
 - Next **N** lines will contain **N** space separated distances between cities, in order.
 - All coordinates and distances will be floating point numbers.
- **Submission** - You are required to submit the following.
 - Code (C, C++, Python, Java). The program should read from *stdin* and write to *stdout*.
 - The output should be tours as space separated indices of cities, per line. Do not write the origin city's index at last again. Rotated tours will be considered the same. Invalid tours will be considered as no tours at all.
 - A file containing the command(s) to be run on terminal to generate any executable or other files required by your program to generate the tours.
E.g. Your submission may contain a makefile to generate an executable so this file should contain the command *make* in a single line. Any more such commands required should be added in new lines.
Name this file "*pre*".
 - Another file containing a single command to run your code.
E.g. If you made an executable in the previous step named *tsp* then this file should contain *./tsp* in a single line, or if you made a python3 code named *tsp.py* then this file should contain *python3 tsp.py* in a single line.
Name this file "*run*".
 - Assume that these commands will be run in the root folder of your zip (unzipped). This is for keeping paths relative to it.
 - A brief report stating your methodology and iterative improvements. (*report.pdf*)
 - The *time limit* for running your code is 300s, after which your process will be terminated. Make sure to print your best tours to stdout as soon as you find them because only the last valid tour will be considered for evaluation.
 - Do not use multi-threading for this assignment.
 - Submit a *zip* (with *report.pdf*, *pre* & *run* in root) with your groupID as it's name. E.g. 1.zip

Evaluation

- You will be evaluated on the basis of the cost of your tours.
- We are providing a few test cases so that you can evaluate your performance and improve before final submission.
- Judgement will be made in a relative fashion with some thresholding.
- We are also attaching results from previous batches so that you can compare your results.
- Maximum points are 10.
- There will also be a public problem (bigger in magnitude) for which the best tour will be awarded 2 extra points. If more than one team gets the same best tour, the points will be split accordingly. You can use as much time or resources as you want to get the best tour for this problem.

Deadlines

Wednesday, 20th Feb 2019

- Trial run - 23:59, ~~Thursday, 27th September, 2018~~ **Note:** This is as important as the final submission as failing to submit here will earn you -3 points. Results of this will be reverted back to you in a couple of days. *Wednesday, 6th Mar 2019*
- Final submission - 23:59, ~~Thursday, 27th September, 2018~~. If your submission fails to run, no excuses will be entertained after this time.
- Bigger Problem - 23:59, ~~Friday, 28th September, 2018~~. Will be released on 1st.

~~_____ (2018-18-18)~~