

CS6380 : Artificial Intelligence  
Assignment 1 (MM: 05)  
Due: Trial run - Sep 7, 2019

## Problem Statement

Travelling Salesman Problem: Given a set of cities and distance between every pair of cities, find the shortest possible route that visits each city exactly once and returns to the starting point.

## Specifications

- **Input:** Your script should read from stdin in the following format.
  - First line will contain either euclidean or non euclidean 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: distance of the nth city from each city in the order.
  - All coordinates and distances will be floating point numbers.
- **Output:** The output should be tours (one tour/line) as *space separated indices of cities*. Do not write the origin city's index in the end again. Invalid tours will be considered as no tours at all.
- **Submission format:**
  - You can write your script in C++, Python, Java. It should read from stdin (in the input format as given above) and write to stdout (in the output format as given above).
  - Your submission should contain the following files in the root folder:
    - Your script
    - Makefile: to generate an executable (not required for a python script)
    - a file named *run*: containing a single command to run your code. For example, If your Makefile contains the commands to generate an executable named tsp then this file should contain *./tsp* in a single line, or if you submitted a python3 code named tsp.py then this file should contain *python3 tsp.py* in a single line.
    - Assume that these commands will be run in the root folder. This is for keeping paths relative to it.
    - Report: A brief report stating your methodology and iterative improvements. Name it <groupID>Report.pdf. Name your root folder as <groupID> and submit it zipped as <groupID>.zip (find a sample submission attached)
    - The time limit for running your code is 300s, after which your process will be terminated. Make sure to print your best tours (only) to stdout as soon as you find them because only the last valid tour will be considered for

evaluation, which also means you should print at least one valid tour before timeout.

### **Evaluation**

- You will be evaluated on the basis of the cost of the tours. As stated above, your script should write your best tours to stdout. Our script will pick the last valid tour that your script outputs before the timeout and compute its cost.
- We are providing a few test cases so that you can evaluate your performance and improve before final submission.
- Marks will be awarded in a relative fashion as on how good your tours are as compared to others'.

### **Deadlines**

- **Trial run - 23:55, September 7, 2019**

Trial runs are important as you will get feedback on how good your tours are, as compared to others or at least you would get to know whether your script interfaces properly with our script. Failing to submit here will earn you (-2) points. Results and the feedback of the trial run will be published in a couple of days.

- **Final submission - 23:55, September 14, 2019.**

If your submission fails to run here, you get zero in this assignment.