

# CS6380 : Artificial Intelligence

## Assignment 1

Due: Trial Run- 16 October 2020

### Problem Statement

**Travelling Salesman Problem:** Given a set of cities and distance between every pair of cities, find the shortest possible route that visits every 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 **<ROLL\_NO>\_Report.pdf**. Name your root folder as **<ROLL\_NO>** and submit it zipped as **<ROLL\_NO>.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 sample test cases so that you can evaluate your performance and improve before final submission. Please note that we will be using different test cases for evaluation.
- Marks will be awarded in a relative fashion as on how good your tours are as compared to others.

## Deadlines

- **Trial Run - 23:55 16 October, 2020**

Note: This is important as you will get a feedback on how your script is performing or/and at least you would get to know the bugs in your script, if any (remember the I/P - O/P specification?).

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 23 October, 2020**

If your submission fails to run here, no excuses will be entertained further.