



## Homework 5

tpetit@wpi.edu

*This is an individual assignment, graded over 100 points*

**For the homework, upload on the course website an archive file FIRSTNAME.LASTNAME.HW5.zip (replace FIRSTNAME and LASTNAME by your first and last names).**

Create your own application! You are 100% free to design the program as you wish, i.e., determine the classes, methods, functions, etc. The only constraints are to use the provided datasets and to provide a program that meets the specifications. Only free Python **packages** are allowed (no need to pay for executing your code). Your code will be evaluated with the following criteria:

1. The program works.
2. The program meets the specifications.
3. The program is user-friendly, easy to use and potential crashes are caught.
4. The code is clear, well structured. Random values are generated with seeds.
5. Good compromise between the quality of the solution and the execution time.
6. Extra features (optional but may give some extra points).

The goal is to make a program that solves traveling salesman problem: A Salesman wishes to travel around a given set of cities, and return to the beginning, covering the smallest total distance. **Each city must be visited once.**



Traveling Salesman Problem (TSP): A Salesman wishes to travel around a given set of cities, and return to the beginning, covering the smallest total distance



Worcester Polytechnic Institute

The goal is to implement a program that uses a data file of city coordinates, randomly extracts  $n$  cities from that data file ( $n$  should be a parameter of your program), and calculates a tour. The program should try to minimize the total distance of the tour (which includes the return to the first city) as much as possible.

## Week 12

- Explore the data: `uscities.csv`. Three columns are essential: city names, latitude and longitude (city coordinates).
- Perform your own research to determine how you can approximate the distance between cities from their coordinates. You may use the `mpu` package.
- Design the structure of your program: How will you represent a tour in your program? Will you use classes or functions? which ones? how will you code the data in your program? How will you randomly extract some cities? etc.

Sources: Python documentation, pandas documentation, web search for distance approximation. Ask questions!

### Week 13

- Implement the classes/functions, except the algorithm for computing the tour.
- Implement a greedy algorithm for computing the tour.
- Test your program extensively.

Sources: Week 12 and 13 materials.

### Week 14

- Write a short README.txt file that explains how to launch your program. If you used external packages, include all the explanations for installing them.
- (*optional*) Add additional features, such as a pyplot window that displays the solution (the tour), or a local search algorithm that gives better solutions than the greedy algorithm.
- Create a .zip archive with all the Python files and the README.txt and submit in on Canvas before the due date. Only .zip files, no other formats!

Sources: Week 12 and 13 materials.