Assignment #4: Constraint Satisfaction Problem (CSP)

**Course Instructor:** Dr. Shahnawaz Qureshi  
**Course:** Artificial Intelligence

**Due:** 20th November 2023, Monday 11:59 p.m.

**Instructions:**

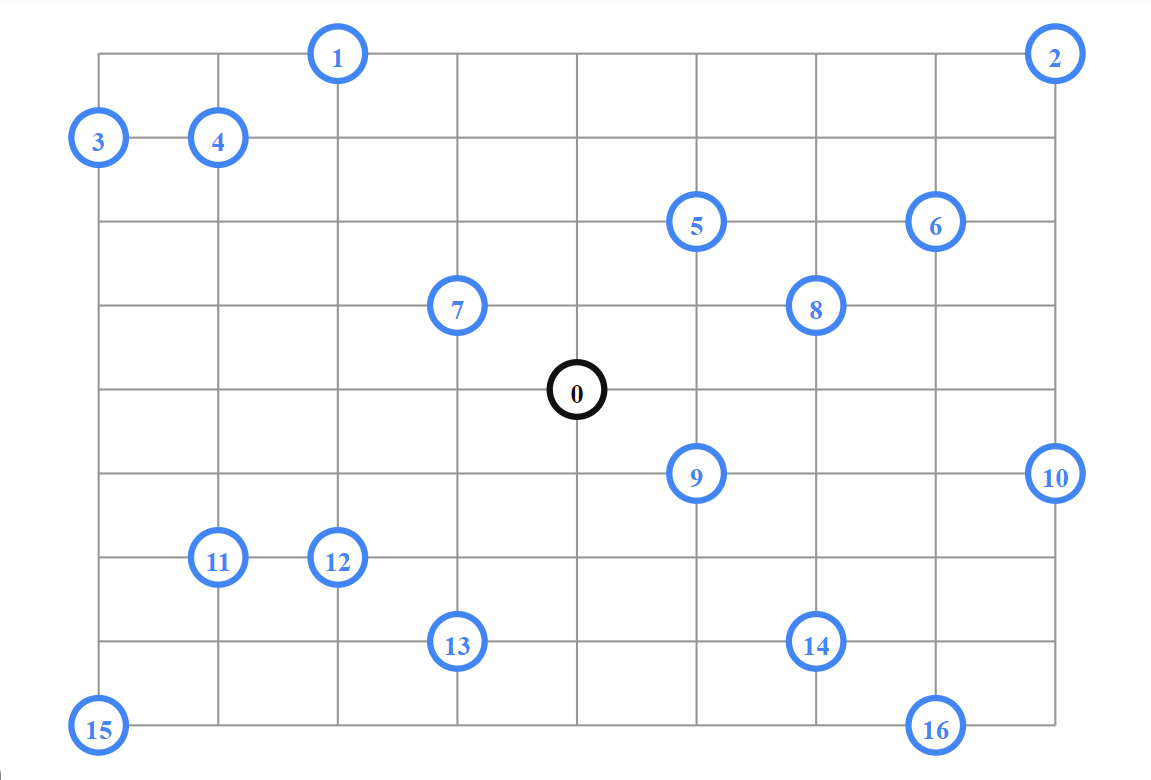
* You are allowed to discuss this assignment verbally but a plagiarism check will be maintained.
* If there are any assignment related queries, please first watch the video posted along with this document. If your query is still unresolved, you can e-mail one of the TA’s.
* Submit your file under the naming **convention iXXXXXX\_Section.ipynb (E.g. i210328\_A.ipynb).**
* Do not zip your files. A 5% penalty will be applied :(.
* Submit the assignment on time.
* Late submissions will receive a 10% penalty.
* Assignment will be due by **20th November 2023, Monday 11:59 p.m.**
* Deadline will not be extended.
* Lastly, have fun :). The goal with this assignment to try to solve a real-world optimization problem. I tried to simplify this assignment a bit by ignoring capacity and time factors, using some better constraints and being a bit lax on the optimization requirements. However, if you are able to incorporate capacity and time constraints in this assignment later on, it will be a valuable asset to include on your GitHub profile.

# Assignment #4: Vehicle Routing Problem (VRP)

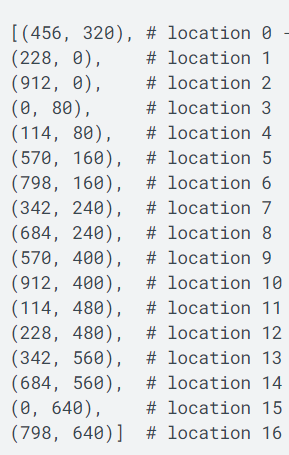
**Introduction:** The goal is to find optimal routes for multiple vehicles visiting a set of locations. (When there's only one vehicle, it reduces to the Traveling Salesperson Problem -which you have covered in Assignment #2). For our scenario, the “optimal route” will refer to the routes with the least total distance for all our vehicles combined.

## Assignment Scenario:

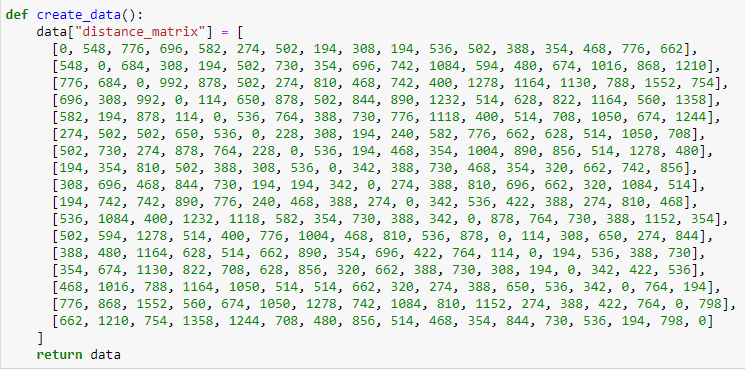
Imagine a company that needs to visit different cities in an area made up of identical rectangular blocks with ‘n’ number of vehicles at their disposal. A diagram of the cities is shown below, with the starting city marked in black and the cities to visit in blue. Your vehicles must start and end at the starting city (City 0).



Here are the locations for every city on the rectangular map. Location 0 would correspond to City 0:

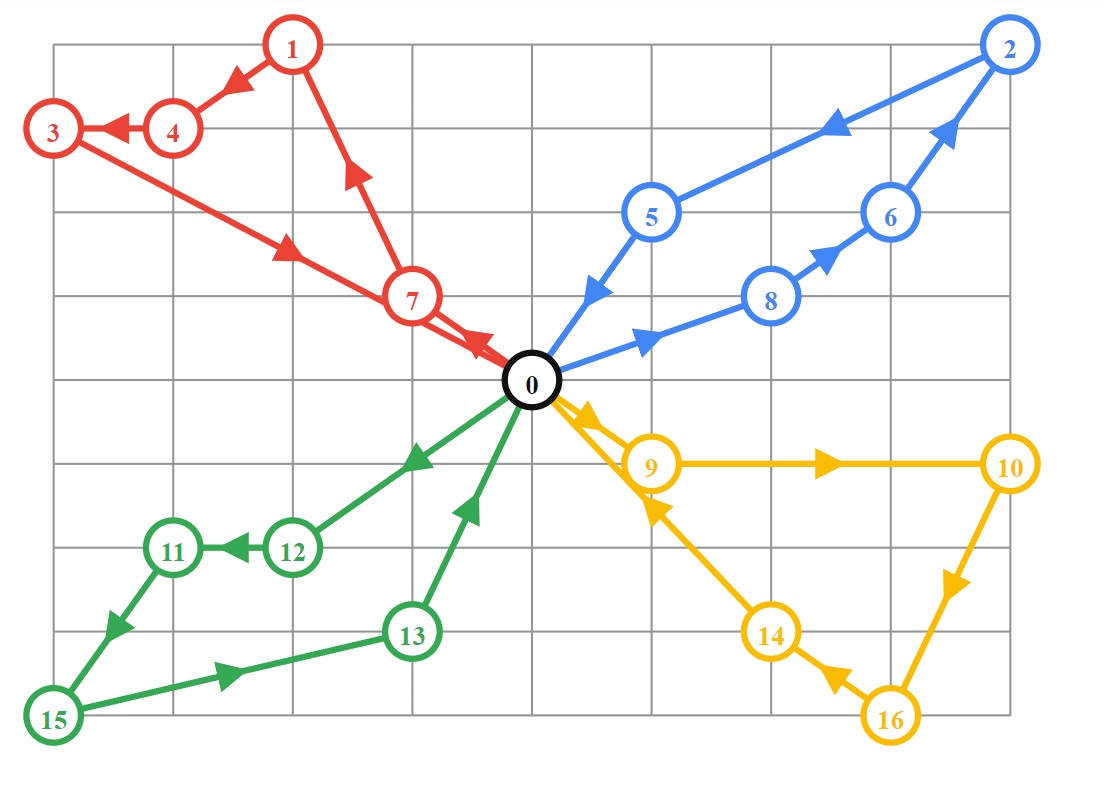


A distance matrix is also given below where the indices of the arrays correspond to the cities labelled in the diagram up above. The distances are calculated according to the Manhattan distance formula using the coordinates from the above picture:



As such, the distance matrix is computed. It is symmetrical and a path does exist between every city.

Now, assuming we have 4 vehicles to dispatch, we have to find the most optimal route for those 4 vehicles which will have the least total distance. As such, you will find this as a solution:



Now, with the basic introduction to the scenario done. Here are the questions you are required to answer:

**Q 1) Write a python program which calculates and shows the most optimal route for ‘n’ vehicles and the overall total distance. A list of city coordinates and its respective distance matrix is provided to you in an .ipynb notebook. You are required to solve this problem whilst following these constraints:**

**Constraint #1:** Each vehicle must start from City 0 and end its route at City 0.

**Constraint #2:** All cities should be visited **exactly** once (Apart from City 0 itself).

**Constraint #3:** You must find ‘n’ routes which will have the leasttotal distance combined. Each route corresponds to a different vehicle.

**Constraint #4:** No libraries apart from numpy and matplotlib are allowed :)

You may use any approach/methodology to solve this problem (i.e. Forward Checking, Arc Consistency, Backtracking search, Brute Force etc.). Due to the difficulty of the problem itself, I’m not that interested on how efficient or optimal your code and methodology is. However, your solution should not get stuck in a local optima and must be sensible.

Your output must be in the form of (Assuming 4 vehicles given as input):

Route 1: 0 -> 7 -> … -> 0.

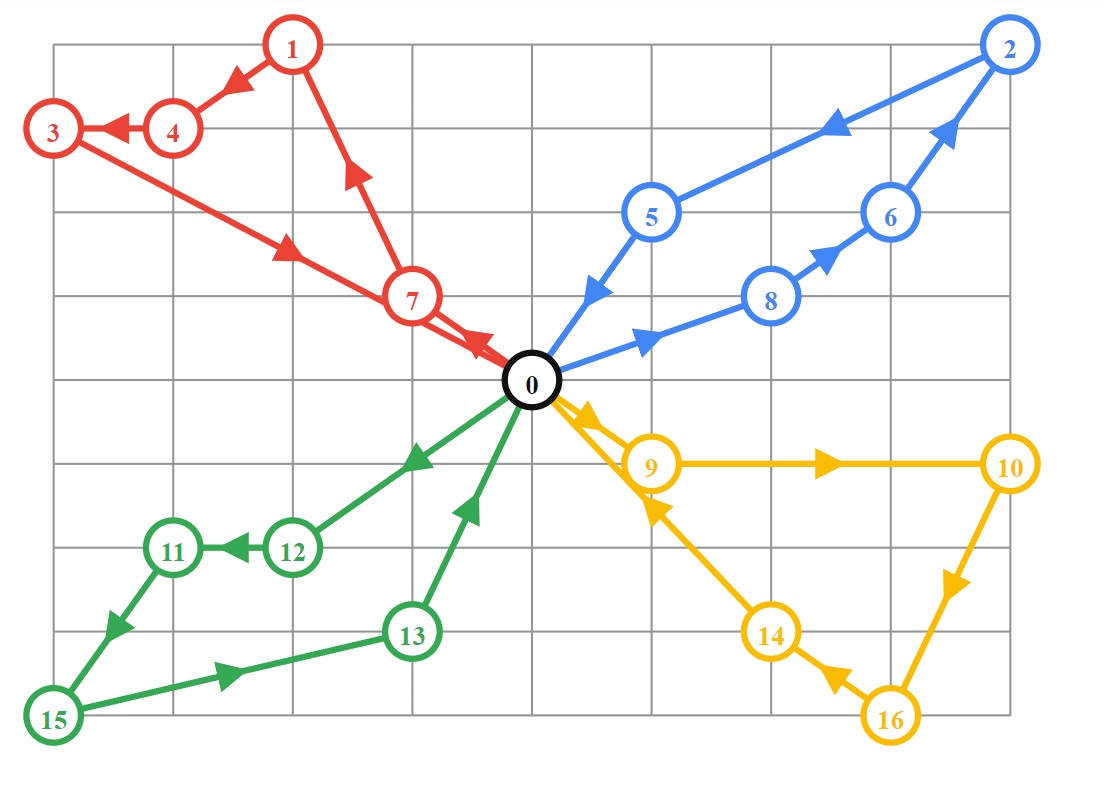
Route 2: 0 -> 8 -> … -> 0.

Route 3: 0 -> 9 -> … -> 0.

Route 4: 0 -> 12 -> … -> 0.

Total Distance: xxxx units

**Q 2) Plot a graph showcasing the route taken by each vehicle from your output in a different color (Similar to something like this below).**



**Rubric:**

|  |  |
| --- | --- |
| Category | Marks |
| Methodology Used | **20** |
| Genericness | **10** |
| Code | **50** |
| Text output | **10** |
| Graph output | **10** |

Total: 100 marks