**Project 1**

**Artificial Intelligence**

Spring 2023

**Distributed: 27 Jan 2023**

**Due Part A: Saturday 4 February 2023**

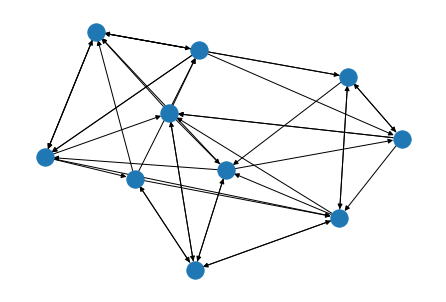
**Part B: Saturday 18 February 2023**

*[Solutions to this assignment must be submitted via CANVAS prior to midnight on the due date. Submissions up to one day late will be penalized 10% and a further 10% will be applied for the next day late. Submissions will* ***not*** *be accepted if more than two days later than the due date.]*

This project may be undertaken either individually or in pairs. Please state the ***names*** of the (person or persons undertaking the project and the ***contributions*** that each has made. Only ONE submission must be made per group.

**Purpose**: To gain a thorough understanding of the working of an agent to assist the agent’s navigation in a stochastic setting that ***mimics a real-world environment***. Thus, this requires reading through the project specification carefully to make sure that you understand its requirements. Jot down any questions/doubts that you may have and feel free to ask me questions in class or in person. Together with your partner, work out a strategy before you start coding the solution in Python. Given the limited timeframe for the project, some simplifications have been applied. In all cases, when simplifications are used, they have been pointed out explicitly. You may find it useful to compile a) a list of all functional requirements and b) a list of all assumptions before starting to code.

**Environment Description:** The application is a ridesharing one to be deployed in any given area of geographical coverage spanning a city. The environment is specified in the form of a graph structure as represented below (simply an example; not the entire road network). The graph that you will use consists of 100 nodes, with an average connectivity of 2 – this means setting your p value at 0.02.



Each node in the graph represents either a potential pickup or drop-off point. The distance between a pair of connected nodes (edge length) is **1** mile. All edges are of the same length. Edges are **bidirectional**. For the purposes of this project, we will assume that vehicles travel at a constant speed of **30 mph**.

All system operations such as *pickup, drop off and van parking can take place at* ***every clock tick* which happens every minute**. The only exception is *pickup scheduling* that occurs **every 4 minutes**.

All *pickups, drop offs and van parking can only occur at nodes, not in-between at edges.*

# Scheduling

## Van Scheduling

The ABC company employs a fleet of vehicles of 30 vans to service the transportation needs of its customers. Two types of scheduling occur. Firstly, the company assigns a vehicle to a customer based on proximity of the vehicle to the customer’s location.

## Customer Pickup Order Scheduling

The second type of scheduling is determining the order of pickup of customers from a queue containing pending requests. The system operates on an internal clock that ticks once every minute. Scheduling of pickup order happens every 4 minutes on the 4th clock tick. Up to 3 customers may be scheduled for pickup depending on the number already in the van. Once a pickup order *is determined it will never be changed at a future point even if a new customer request is at a closer point to the van than any of the already scheduled pickups*.

Each customer enters their pickup point (node number in graph) and drop-off point (also a node number) when placing a request.

The order is determined by the distance traveled to service the pickups. Suppose for example there is 1 passenger in the van and 5 pending requests are in the queue. Since there is space for 2 more, the first two requests A and B in the queue (inserted in time arrival order) are removed and examined in terms of their pickup and drop off locations. If picking up B first and then proceeding to A’s pickup location is shorter than first proceeding to A’s location, then the pickup order will be B first and A second. Likewise, it will be necessary to determine the pickup order for 3 customers if they are scheduled together at the same time.

We will also assume that cancellations are ***not*** made by either the customer or the company once a reservation has been made.

## Customer Drop off Order Scheduling

Drop offs, just like pickups are scheduled on the basis of distance to be traveled. Drop offs can occur at any time that there are passengers in the van. Thus, for example, if two requests are serviced to pick up A and B, then B could be picked up first and then dropped off before A’s pickup is scheduled. Thus, pickups can be interspersed with drop offs – i.e., drop offs can take place in between pickups.

## Van Idling

If a van is not assigned a new passenger after dropping off its last one, then ***it simply parks at the drop off point and waits for the next passenger pickup request***. We will assume that parking is always available.

## Project Requirements

### Part A (due one week after project is handed out)

Your task in this project is to implement the following requirements:

R1. Produce a pseudo code version of the algorithm needed for scheduling customer pickups.

(**15 marks)**

R2. Produce a pseudo code version of the algorithm needed to plan a route for passenger drop offs.

**(15 marks)**

**Note:** Pseudo code is a high-level representation of actual code and so do not submit actual code. R1 and R2 are meant as thinking exercises prior to coding. All pseudo code must be typewritten and submitted in a pdf document. No handwritten versions will be accepted.

### Part B (due two weeks after Part A is submitted)

R3 Implement a Python program that incorporates the two algorithms in R1 and R2 above.

**(30 marks)**

R4. Assume that the rideshare is provided in an 8-hour timespan in which bookings are accepted; that the company operates a fleet of 30 vehicles and that 450 to 600 reservations are made per hour (of simulation time) across the entire fleet**,** what is:

1. the average distance traveled (over the fleet) per day when run a road network of 200 nodes and average connectivity of 2? **(5 marks)**
2. the average number of trips (over the fleet) per day when run a road network of 200 nodes and average connectivity of 2? **(5 marks)**

Note that reservations occur at random points in the clock and may originate at random nodes in the graph. Likewise, destinations are also random.

R5. Now vary the fleet size to 60 vehicles and suppose that reservations occur per hour with the same booking rate mentioned in R4 above. What are the new values of average distance and average number of trips traveled per day taken across the entire fleet? **(10 marks)**

R6 Now run your code across a new graph having a connectivity of 4 instead of 2 whilst still having 200 nodes - an example is shown below. What is the new value of the average distance when the booking rate given in R4 is applied? Why does it differ? **(20 marks, 10 for value, 10 for reason)**

Hand in:

1. A short report (between 1 and 2 pages in size) describing the approach that you took. In addition to the description of your approach you should answer the 6 requirements given above. If working in a team of size 2, a clear demarcation of contribution between team members is required.
2. Any additional assumptions that you made while undertaking this project.
3. The complete set of Python code that you used. This needs to be submitted both as a pdf document (no images please) as well as your original Google Colab notebook. Note that you do not have to submit the notebook itself; a publicly available link (make sure you check this) to it is required.
4. A pdf document (not an image) of all your Python code contained in your Colab notebook.