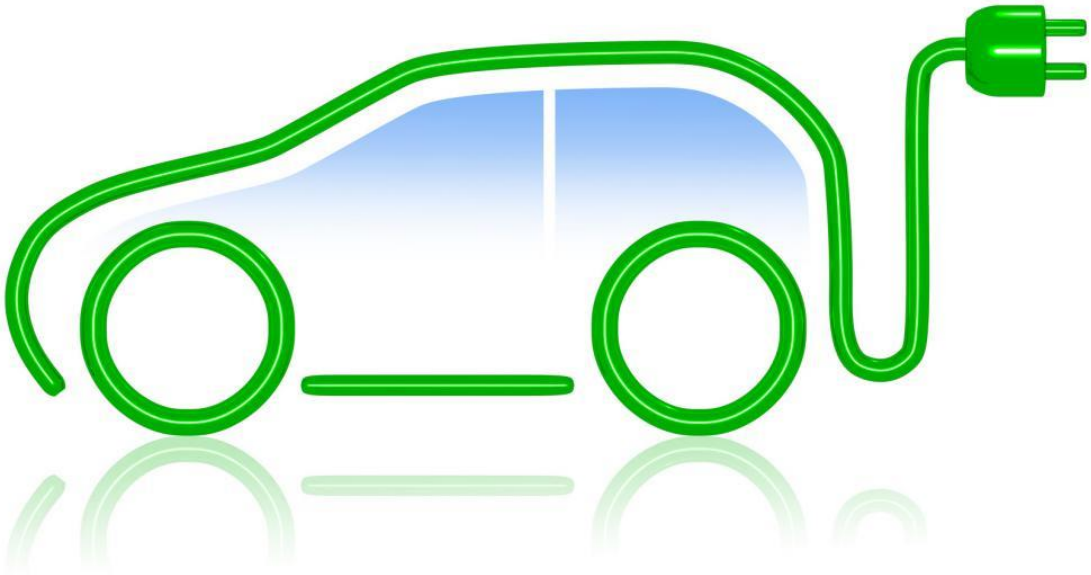


# AIFA Assignment - 1

3. ELECTRIC VEHICLES

Bimal Kumar Sahoo | AIFA (AI61005)

GROUP MEMBERS:

- 17IM30032 – BIMAL KUMAR SAHOO
- 17IM30013 – VIVEK VANAD NARAYANE
- 17IM30012 – JITENDER SWAMI
- 17IM10028 – VIKASH KUMAR

# PROBLEM STATEMENT:

**3. [Electric vehicles]** Consider a city network where we need to route a set of electric vehicles which may require to be charged during its journey from some source to some destination. Let us assume that we have $n$ cities ($v_1; v_2; \ldots ; v_n$) and the distance between cities $v_i$ and $v_j$ be $e_{ij}$ (if two cities are not connected directly then $e_{ij} = 1$ and $e_{ij} = e_{ji}$): Assume that each city has a single charging station which can charge one EV at a time. Consider a set of $k$ EVs namely $P_1; P_2; \ldots ; P_k$: For each EV the following information is provided -

(a) $S_r$ - source node
(b) $D_r$ - destination node
(c) $B_r$ - battery charge status initially
(d) $c_r$ - charging rate for battery at a charging station (energy per unit time)
(e) $d_r$ - discharging rate of battery while traveling (distance travel per unit charge)
(f) $M_r$ - maximum battery capacity
(g) $s_r$ - average traveling speed (distance per unit time).

Assume that all vehicles start their journey at $t = 0$ and $P_r$ reaches its destination at $t = T_r$. We need to route all the vehicles from their respective sources to destinations such that max $\{T_r\}$ is minimized. You need to develop both optimal as well as heuristic algorithms.

# INTRODUCTION:

Distance matrix:

- Represents the edges of each corresponding node.
- Is a symmetric matrix with ($e_{ij} = e_{ji}$).
- Is 'inf' if the nodes are not connected.

Car's data:

(a) $S_r$ - source node
(b) $D_r$ - destination node
(c) $B_r$ - battery charge status initially
(d) $c_r$ - charging rate for battery at a charging station (energy per unit time)
(e) $d_r$ - discharging rate of battery while traveling (distance travel per unit charge)
(f) $M_r$ - maximum battery capacity
(g) $s_r$ - average traveling speed (distance per unit time).

N – cities

- Each with a single charging station which can charge one vehicle at a time.

K – vehicles

- Have to reach to the destination with the least possible time.
- Have an option to charge at any node (at once, in case of multiple vehicles).

# FORMULATION:

**Objective**:

MINIMISE max{$T_r$}

**States**:

State of each vehicle is the current battery charge, current node, current queue in the charging station, destination node, maximum battery capacity.

**Initial state:**

Vehicle start from the source node with current battery charge as the initial charge. (Source node for different vehicles could be different.)

**Goal Sate:**

State when the current node is same as the destination node.

**Constraints**:

- Battery charge cannot exceed Maximum battery capacity for the respective vehicle.
- If two nodes are not connected, the vehicles cannot travel through the edge. (distance = inf)
- Each node has a single charging station.
- One vehicle can be charged at a time.
- If there is one vehicle in a node, the queue waiting length is zero.

# ALGORITHM USED:

## A* algorithm:

A* is a sophisticated search algorithm, also known as a best-first search since it is written in terms of weighted graphs. The goal is to find the shortest path from a given starting node to a given goal node (least distance traveled, shortest time, etc.). It accomplishes this by keeping track of a tree of paths that begin at the start node and extending those paths one edge at a time until the termination requirement is met.

A* must decide which of its paths to extend at each iteration of its main loop. It does so use the path's cost as well as an estimation of the cost of extending the path all the way to the target. A*, in particular, chooses the shortest path.

$f(n) = g(n) + h(n)$

where, *h(n) is a heuristic function that calculates the cost of the cheapest path from n to the destination, where n is the next node on the path, g(n) is the cost of the path from the start node to 'n', and g(n) is the cost of the path from the start node to 'n'.*

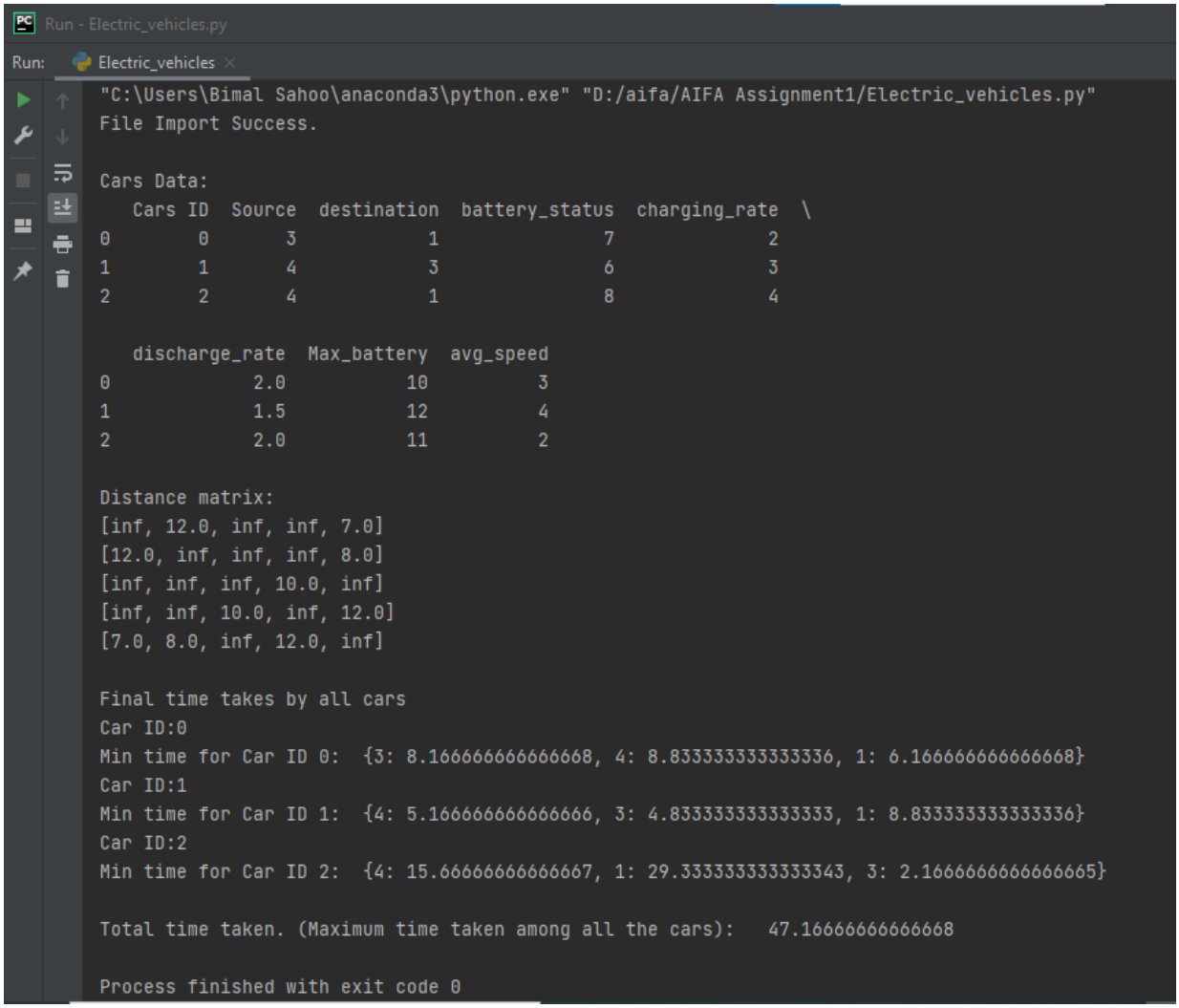
# APPROACH:

- In the given problem, g(n) represents the time consumed till the vehicle reaches the node 'n'.
- For each node, the shortest route is calculated for reaching the destination. The h(n) is then taken as the expected distance to cover divided by the average speed of the vehicle for the corresponding node n.
- This process is continued till the goal state is reached, i.e., all the vehicles have reached the destination node.
- The output time is the maximum time taken by any of the vehicle.

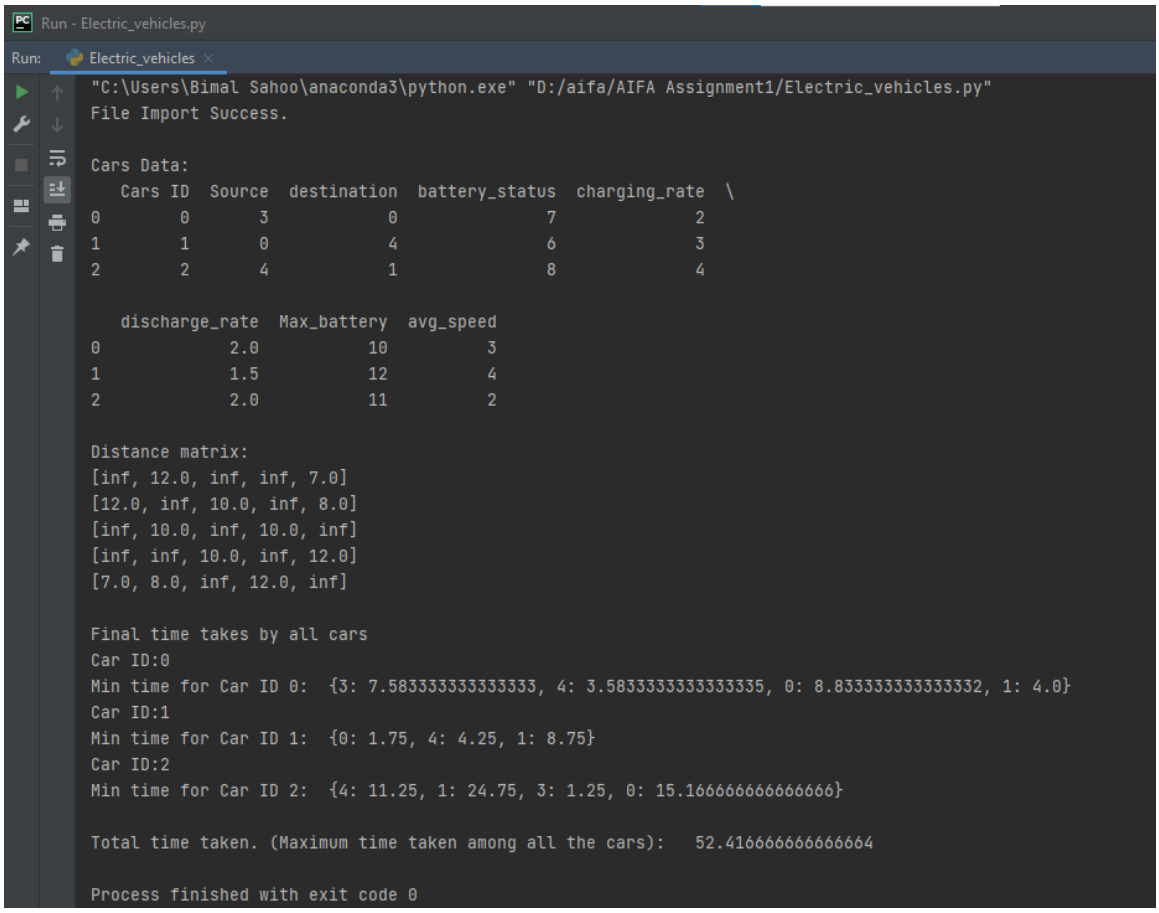
# OUTPUT:

Below is the output for the given test case inputs.

Test case 1:



Test case 2:


Run - Electric_vehicles.py
Run:
Electric_vehicles
"C:\Users\Bimal Sahoo\anaconda3\python.exe" "D:/aifa/AIFA Assignment1/Electric_vehicles.py"
File Import Success.
Cars Data:
Cars ID Source destination battery_status charging_rate \
0 0 3 0 7 2
1 1 0 4 6 3
2 2 4 1 8 4
discharge_rate Max_battery avg_speed
0 2.0 10 3
1 1.5 12 4
2 2.0 11 2
Distance matrix:
[inf, 12.0, inf, inf, 7.0]
[12.0, inf, 10.0, inf, 8.0]
[inf, 10.0, inf, 10.0, inf]
[inf, inf, 10.0, inf, 12.0]
[7.0, 8.0, inf, 12.0, inf]
Final time takes by all cars
Car ID:0
Min time for Car ID 0: {3: 7.583333333333333, 4: 3.5833333333333335, 0: 8.833333333333332, 1: 4.0}
Car ID:1
Min time for Car ID 1: {0: 1.75, 4: 4.25, 1: 8.75}
Car ID:2
Min time for Car ID 2: {4: 11.25, 1: 24.75, 3: 1.25, 0: 15.166666666666666}
Total time taken. (Maximum time taken among all the cars): 52.416666666666664
Process finished with exit code 0