AIFA ASSIGNMENT SPRING 2021

ELECTRIC VEHICLE ROUTING PROBLEM

USER GUIDE

TABLE OF CONTENTS

STATEMENT OF SCOPE

The major inputs are

1. Number of cities
2. Number of EVs and their starting and destination cities for each
3. The average speed of EVs.
4. The discharge rate of EVs each.
5. Charging rate of EVs at a city each.
6. Maximum battery capacity for each EV.
7. Initial Battery Status
8. Graph or network of cities and distance between them.

We will process all the values and use a pathfinding algorithm. The output gives the time taken for each EV to reach its destination.

TERMINOLOGIES AND DEFINITION

Evs: It is a dictionary of details of the cars where keys are the cars {P1, P2, P3…,Pn}
Where **Pi** is itself a dictionary and contains the values like below example

```
{
    'P1' : {
        'source' : 'V1',
        'destination': 'V8',
        'init_battery' : 5,
        'charging' : 2,
        'discharging' : 3,
        'max' : 100,
        'speed' : 12
    },
```

Graph: It is a dictionary that contains information of every city(node) and the cities which it is connected to and the distance between them. The keys of the graph are the cities{V1,V2,...Vr}. For example

```
{
    'V1' : {
        'V2' : 8,
        'V4' : 4,
        'V5' : 10
    }
```

Cars: It is a list of EVs i.e [P1, P2, P3...Pn]
Cities: It is a list of Cities i.e [V1, V2, V3...Vr]
max_dist: It is the maximum distance that a car can travel with a maximum battery level.
distances: It is the dictionary that stores the shortest distance between every source city to every destination city via optimal path..
parents: It is a dictionary that contains parent cities of every city in the optimal path.
iterate: It is a list of unvisited cities.
mincity: it contains the cities which to be next visited,(its cities are going to be opened next and that city is removed from the iterate list).
upd: it contains the updated distance.
pathlist: it is a list that contains the optimal path from source node to destination city. Its elements are the cities that come through in the path.

SYSTEM REQUIREMENTS

We have implemented our algorithm in python notebook (.ipynb) filetype through Jupyter notebook and Google Collab.
This type of file requires Python version 3.3 or greater version installed in your system.

1. You can download the file from GitHub and run it in Jupyter notebooks.
2. You can alternatively run the file in Google Collab via uploading the .ipynb file.

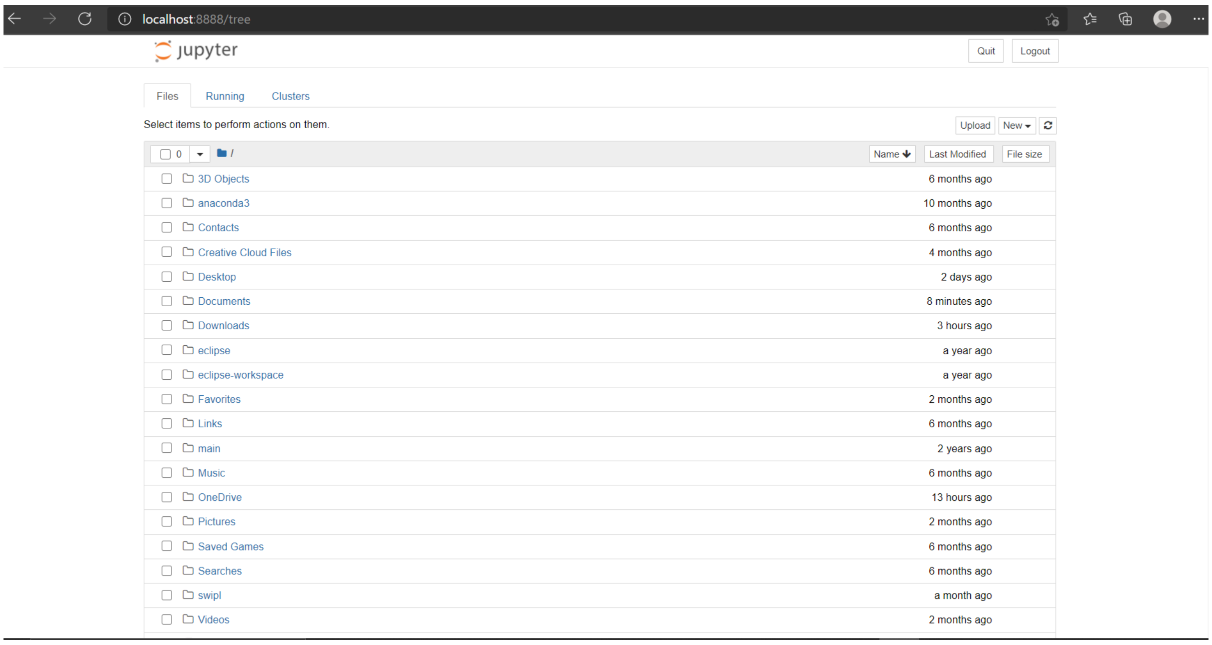
HOW TO RUN THE FILE

To run the .ipynb file in Jupyter notebook

Follow the steps

1. **Download the file from github repository.**

2. **Open Jupyter notebooks in your system.**



localhost:8888/tree

jupyter | Quit | Logout

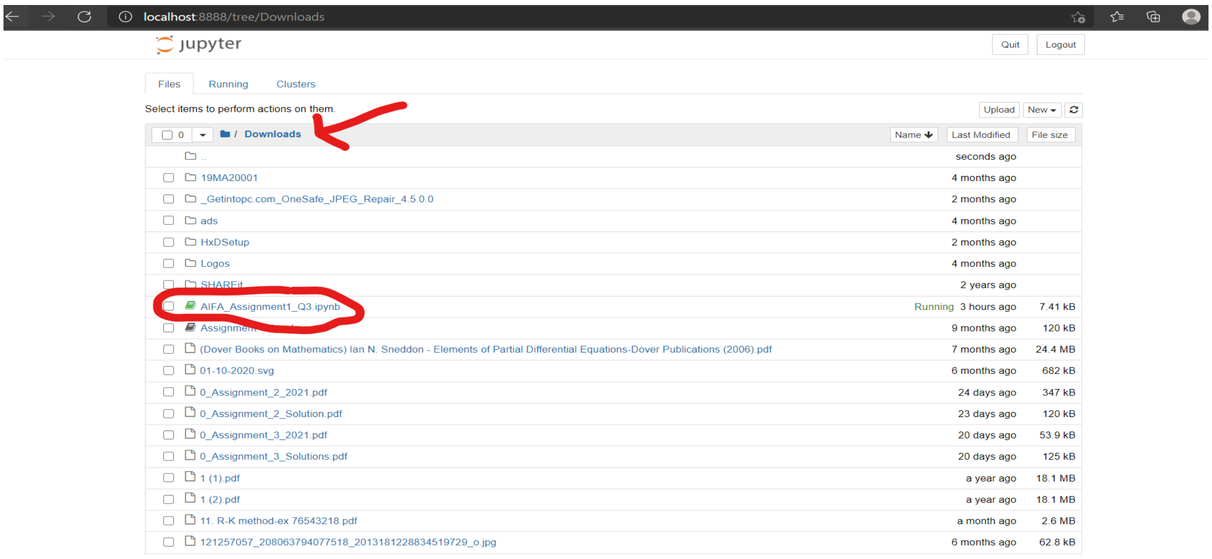
Files | Running | Clusters

Select items to perform actions on them.

Upload | New | Name | Last Modified | File size

Name	Last Modified	File size
3D Objects	6 months ago	
anaconda3	10 months ago	
Contacts	6 months ago	
Creative Cloud Files	4 months ago	
Desktop	2 days ago	
Documents	8 minutes ago	
Downloads	3 hours ago	
eclipse	a year ago	
eclipse-workspace	a year ago	
Favorites	2 months ago	
Links	6 months ago	
main	2 years ago	
Music	6 months ago	
OneDrive	13 hours ago	
Pictures	2 months ago	
Saved Games	6 months ago	
Searches	6 months ago	
swipl	a month ago	
Videos	2 months ago	

3. **Open the location where the downloaded file is stored.Here is a screenshot of the location in my system.**



localhost:8888/tree/Downloads

jupyter | Quit | Logout

Files | Running | Clusters

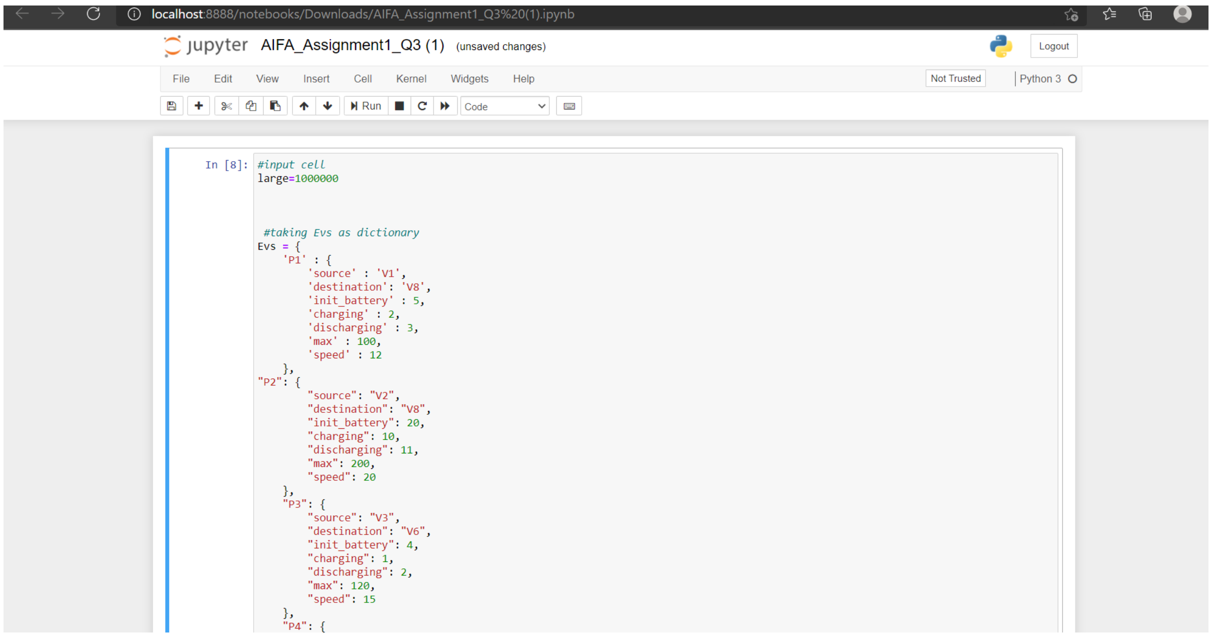
Select items to perform actions on them.

Upload | New

/ Downloads

Name	Last Modified	File size
..	seconds ago	
19MA20001	4 months ago	
_Getintopc.com_OneSafe_JPEG_Repair_4.5.0.0	2 months ago	
ads	4 months ago	
HxDSetup	2 months ago	
Logos	4 months ago	
SHAREit	2 years ago	
AIFA_Assignment1_Q3.ipynb	Running 3 hours ago	7.41 kB
Assignmen[illegible]	9 months ago	120 kB
(Dover Books on Mathematics) Ian N. Sneddon - Elements of Partial Differential Equations-Dover Publications (2006).pdf	7 months ago	24.4 MB
01-10-2020.svg	6 months ago	682 kB
0_Assignment_2_2021.pdf	24 days ago	347 kB
0_Assignment_2_Solution.pdf	23 days ago	120 kB
0_Assignment_3_2021.pdf	20 days ago	53.9 kB
0_Assignment_3_Solutions.pdf	20 days ago	125 kB
1 (1).pdf	a year ago	18.1 MB
1 (2).pdf	a year ago	18.1 MB
11. R-K method-ex 76543218.pdf	a month ago	2.6 MB
121257057_208063794077518_2013181228834519729_o.jpg	6 months ago	62.8 kB

4. **Open the file by clicking on it. Here is a screenshot of the opened file.**

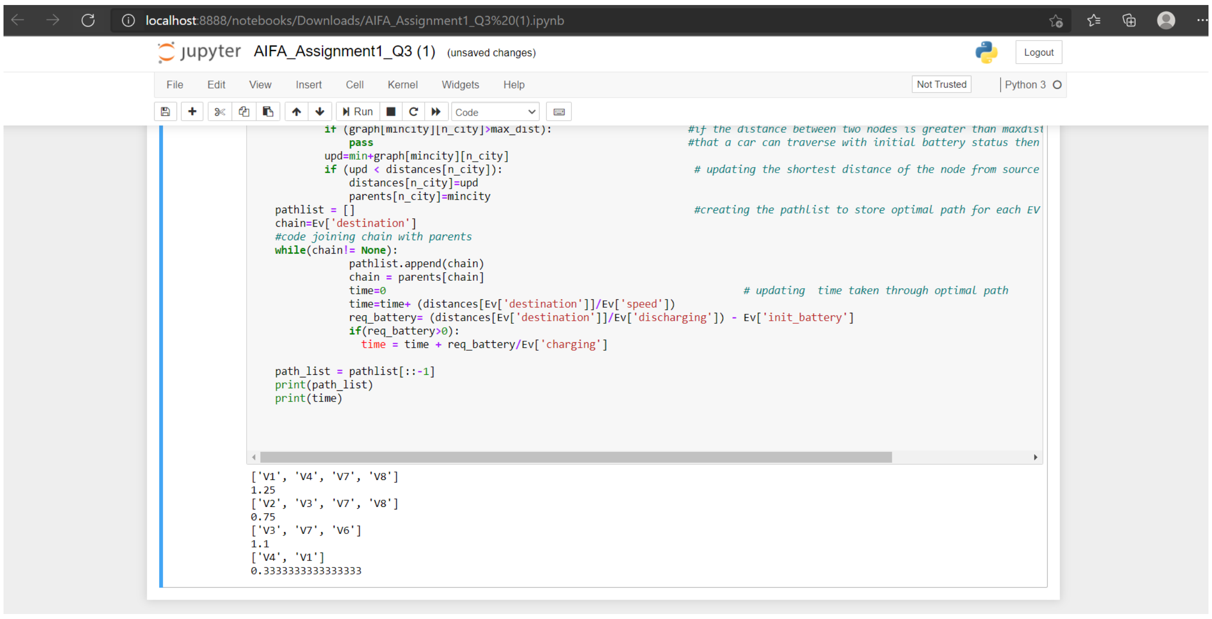

localhost:8888/notebooks/Downloads/AIFA_Assignment1_Q3%20(1).ipynb
jupyter AIFA_Assignment1_Q3 (1) (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Code

```
In [8]: #input cell
        large=1000000



         #taking Evs as dictionary
        Evs = {
            'P1' : {
                'source' : 'V1',
                'destination': 'V8',
                'init_battery' : 5,
                'charging' : 2,
                'discharging' : 3,
                'max' : 100,
                'speed' : 12
            },
        "P2": {
                "source": "V2",
                "destination": "V8",
                "init_battery": 20,
                "charging": 10,
                "discharging": 11,
                "max": 200,
                "speed": 20
            },
            "P3": {
                "source": "V3",
                "destination": "V6",
                "init_battery": 4,
                "charging": 1,
                "discharging": 2,
                "max": 120,
                "speed": 15
            },
            "P4": {
```

5. **Now run the input cell and then the function cell.**
6. **There are two cells - input cell and function cell. You can give input values in the input cell as your wish and then again run both the cells one by one.**
7. **The output will be printed as shown below. The path of each vehicle and the time taken will be displayed respectively.**


localhost:8888/notebooks/Downloads/AIFA_Assignment1_Q3%20(1).ipynb
jupyter AIFA_Assignment1_Q3 (1) (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Code

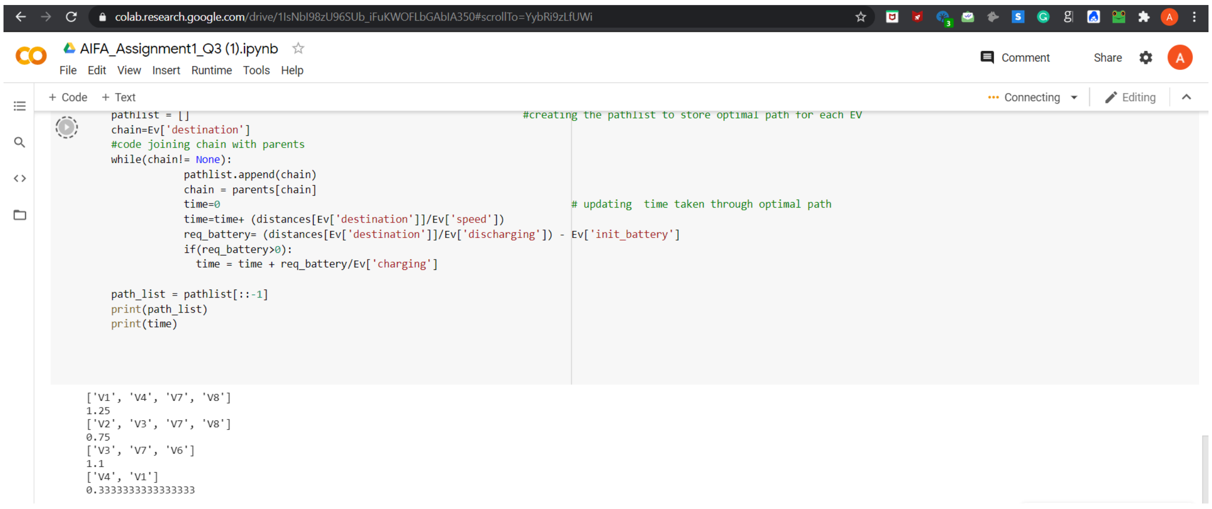
```
            if (graph[mincity][n_city]>max_dist):                    #if the distance between two nodes is greater than maxdist
                pass                                                 #that a car can traverse with initial battery status then
            upd=min+graph[mincity][n_city]
            if (upd < distances[n_city]):                             # updating the shortest distance of the node from source
                distances[n_city]=upd
                parents[n_city]=mincity
    pathlist = []                                                     #creating the pathlist to store optimal path for each EV
    chain=Ev['destination']
    #code joining chain with parents
    while(chain!= None):
                pathlist.append(chain)
                chain = parents[chain]
                time=0                                                     # updating  time taken through optimal path
                time=time+ (distances[Ev['destination']]/Ev['speed'])
                req_battery= (distances[Ev['destination']]/Ev['discharging']) - Ev['init_battery']
                if(req_battery>0):
                  time = time + req_battery/Ev['charging']

    path_list = pathlist[::-1]
    print(path_list)
    print(time)
```

```
['V1', 'V4', 'V7', 'V8']
1.25
['V2', 'V3', 'V7', 'V8']
0.75
['V3', 'V7', 'V6']
1.1
['V4', 'V1']
0.3333333333333333
```

To run the file in Google Collab

1. **Open Google collab.**
2. **Open your file through your system in the google collab.**
3. **Again you can input values in the input cell.**
4. **Run the input cell and function cell.**
5. **The output will be printed as shown below. The path of each vehicle and the time taken will be displayed respectively.**


colab.research.google.com/drive/1lsNbl98zU96SUb_iFuKWOFLbGAbIA350#scrollTo=YybRi9zLfUWi
AIFA_Assignment1_Q3 (1).ipynb
File Edit View Insert Runtime Tools Help
Comment
Share
+ Code + Text
Connecting
Editing

```
    pathlist = []                                        #creating the pathlist to store optimal path for each EV
    chain=Ev['destination']
    #code joining chain with parents
    while(chain!= None):
              pathlist.append(chain)
              chain = parents[chain]
              time=0                                     # updating  time taken through optimal path
              time=time+ (distances[Ev['destination']]/Ev['speed'])
              req_battery= (distances[Ev['destination']]/Ev['discharging']) - Ev['init_battery']
              if(req_battery>0):
                time = time + req_battery/Ev['charging']

    path_list = pathlist[::-1]
    print(path_list)
    print(time)
```

```
['V1', 'V4', 'V7', 'V8']
1.25
['V2', 'V3', 'V7', 'V8']
0.75
['V3', 'V7', 'V6']
1.1
['V4', 'V1']
0.3333333333333333
```

NOTE: Your input should in the dictionary and should be input in the input cell only. In the file, we submitted we have taken some inputs for reference for others to how to give inputs. We have given input of 4 vehicles V1, V2, V3, V4, and their respective properties and 8 cities with details of to which city they are connected.

Conclusion

The system requirements are specified in this document.
Some terminologies and variables are explained in the document.
Statement of scope is also specified.
The filetype in which it is implemented is also specified i.e ipython.
The steps on how to run the file are specified as well.