# AI Assignment 2

**Divyansh Rastogi [2019464]**

## Data Processing:

The cities are divided into two categories, class-1 and class-2.

- class-1 cities can visit all class-1 & class-2 cities.
- class-2 cities can only visit class-1 cities.

The following script is used for data processing:

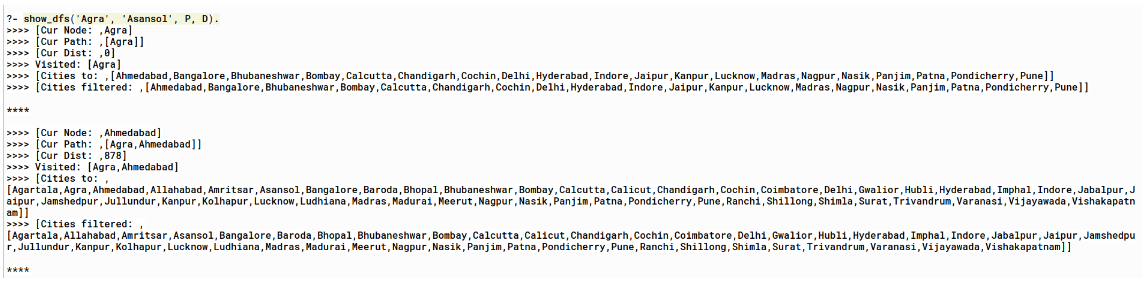
```
import pandas as pd
from tqdm import tqdm

df = pd.read_csv('data.csv')
print(df.head())

# assertions to check validity of data
for i in range(1, len(df.columns)):
    assert (df.columns[i] in df['Distance in Kilometres'].values)
    cities = df['Distance in Kilometres']
    dist = df[df.columns[i]]
    for j in range(len(dist)):
        if (cities[j] not in df.columns):
            continue
        dist_1 = dist[j]
        dist_2 = df.iloc[j][df.columns[i]]
        assert (dist_1 == dist_2)

# seperating class 1 cities in a new dataframe
df_ = {'Distance in Kilometres': list(df.columns[1:])}
for city in df['Distance in Kilometres']:
    df_[city] = []
for from_city in df.columns[1:]:
    for idx in range(len(df[from_city])):
        to_city_dist = df[from_city][idx]
        df_[df['Distance in Kilometres'][idx]].append(to_city_dist)
df_ = pd.DataFrame(df_)

# remove class 1 from class 2
for i in range(1, len(df.columns)):
    city = df.columns[i]
    idx = df['Distance in Kilometres'][df['Distance in Kilometres'] == city].index[0]
    df.drop(idx, inplace=True)

print('Class 1 cities')
print(df_)
print('Class 2 cities')
print(df)

df_.to_csv('class_1_cities.csv', index=False)
```

```
df.to_csv('class_2_cities.csv', index=False)

# ## Heuristic formation

df1_h = df.rename(columns={'Distance in Kilometres': 'Heuristic'}).reset_index(drop=True).copy()
df2_h = df_.rename(columns={'Distance in Kilometres': 'Heuristic'}).reset_index(drop=True).copy()

import requests
def get_geodist(c1, c2):
    URL = f'https://www.distance24.org/route.json?stops={c1}|{c2}'
    r = requests.get(url=URL)
    assert (r.status_code == 200)
    data = r.json()
    return data['distance']

for i in tqdm(range(len(df1_h))):
    city_1 = df1_h['Heuristic'][i]
    for city_2 in df1_h.columns[1:]:
        df1_h[city_2][i] = get_geodist(city_1, city_2)

for i in tqdm(range(len(df2_h))):
    city_1 = df2_h['Heuristic'][i]
    for city_2 in df2_h.columns[1:]:
        df2_h[city_2][i] = get_geodist(city_1, city_2)

df2_h.to_csv('heuristic_1_cities.csv', index=False)
df1_h.to_csv('heuristic_2_cities.csv', index=False)
```

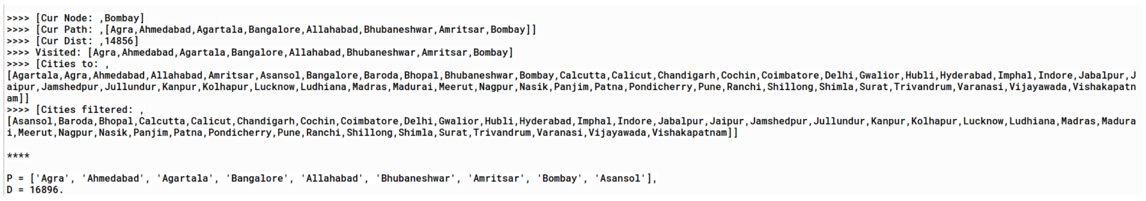
## Search Algorithms:

Each algorithm outputs all intermediate states of the algorithm.

1. Depth First Search



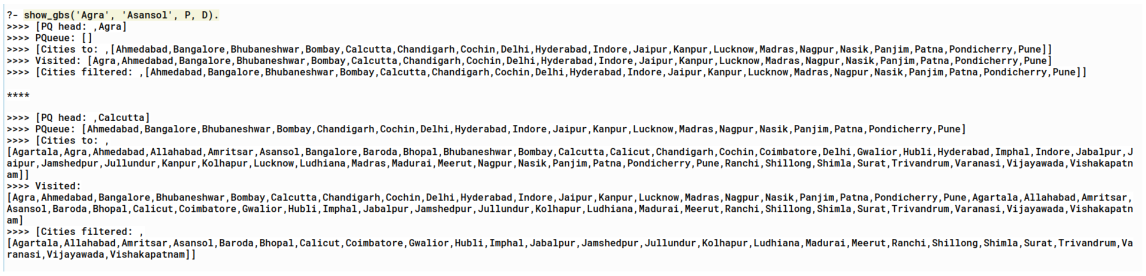
```
?- show_dfs('Agra', 'Asansol', P, D).
>>>> [Cur Node: ,Agra]
>>>> [Cur Path: ,[Agra]]
>>>> [Cur Dist: ,0]
>>>> Visited: [Agra]
>>>> [Cities to: ,[Ahmedabad,Bangalore,Bhubaneshwar,Bombay,Calcutta,Chandigarh,Cochin,Delhi,Hyderabad,Indore,Jaipur,Kanpur,Lucknow,Madras,Nagpur,Nasik,Panjim,Patna,Pondicherry,Pune]]
>>>> [Cities filtered: ,[Ahmedabad,Bangalore,Bhubaneshwar,Bombay,Calcutta,Chandigarh,Cochin,Delhi,Hyderabad,Indore,Jaipur,Kanpur,Lucknow,Madras,Nagpur,Nasik,Panjim,Patna,Pondicherry,Pune]]

****

>>>> [Cur Node: ,Ahmedabad]
>>>> [Cur Path: ,[Agra,Ahmedabad]]
>>>> [Cur Dist: ,878]
>>>> Visited: [Agra,Ahmedabad]
>>>> [Cities to: ,
[Agartala,Agra,Ahmedabad,Allahabad,Amritsar,Asansol,Bangalore,Baroda,Bhopal,Bhubaneshwar,Bombay,Calcutta,Calicut,Chandigarh,Cochin,Coimbatore,Delhi,Gwalior,Hubli,Hyderabad,Imphal,Indore,Jabalpur,J
aipur,Jamshedpur,Jullundur,Kanpur,Kolhapur,Lucknow,Ludhiana,Madras,Madurai,Meerut,Nagpur,Nasik,Panjim,Patna,Pondicherry,Pune,Ranchi,Shillong,Shimla,Surat,Trivandrum,Varanasi,Vijayawada,Vishakapatn
am]]
>>>> [Cities filtered: ,
[Agartala,Allahabad,Amritsar,Asansol,Bangalore,Baroda,Bhopal,Bhubaneshwar,Bombay,Calcutta,Calicut,Chandigarh,Cochin,Coimbatore,Delhi,Gwalior,Hubli,Hyderabad,Imphal,Indore,Jabalpur,Jaipur,Jamshedpu
r,Jullundur,Kanpur,Kolhapur,Lucknow,Ludhiana,Madras,Madurai,Meerut,Nagpur,Nasik,Panjim,Patna,Pondicherry,Pune,Ranchi,Shillong,Shimla,Surat,Trivandrum,Varanasi,Vijayawada,Vishakapatnam]]

****
```



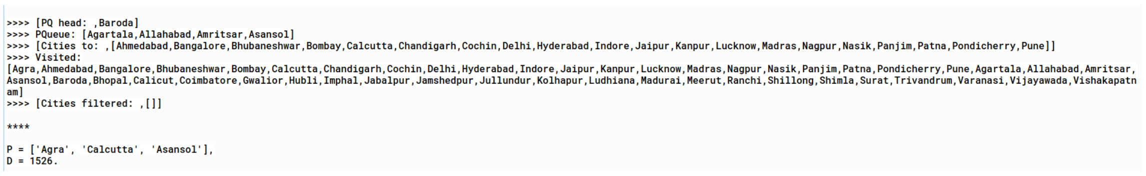
```
>>>> [Cur Node: ,Bombay]
>>>> [Cur Path: ,[Agra,Ahmedabad,Agartala,Bangalore,Allahabad,Bhubaneshwar,Amritsar,Bombay]]
>>>> [Cur Dist: ,14856]
>>>> Visited: [Agra,Ahmedabad,Agartala,Bangalore,Allahabad,Bhubaneshwar,Amritsar,Bombay]
>>>> [Cities to: ,
[Agartala,Agra,Ahmedabad,Allahabad,Amritsar,Asansol,Bangalore,Baroda,Bhopal,Bhubaneshwar,Bombay,Calcutta,Calicut,Chandigarh,Cochin,Coimbatore,Delhi,Gwalior,Hubli,Hyderabad,Imphal,Indore,Jabalpur,J
aipur,Jamshedpur,Jullundur,Kanpur,Kolhapur,Lucknow,Ludhiana,Madras,Madurai,Meerut,Nagpur,Nasik,Panjim,Patna,Pondicherry,Pune,Ranchi,Shillong,Shimla,Surat,Trivandrum,Varanasi,Vijayawada,Vishakapatn
am]]
>>>> [Cities filtered: ,
[Asansol,Baroda,Bhopal,Calcutta,Calicut,Chandigarh,Cochin,Coimbatore,Delhi,Gwalior,Hubli,Hyderabad,Imphal,Indore,Jabalpur,Jaipur,Jamshedpur,Jullundur,Kanpur,Kolhapur,Lucknow,Ludhiana,Madras,Madura
i,Meerut,Nagpur,Nasik,Panjim,Patna,Pondicherry,Pune,Ranchi,Shillong,Shimla,Surat,Trivandrum,Varanasi,Vijayawada,Vishakapatnam]]

****

P = ['Agra', 'Ahmedabad', 'Agartala', 'Bangalore', 'Allahabad', 'Bhubaneshwar', 'Amritsar', 'Bombay', 'Asansol'],
D = 16896.
```

2. Greedy Best First Search

- Heurisitic utilized is the air distance between two cities (fetched using distance24 api).

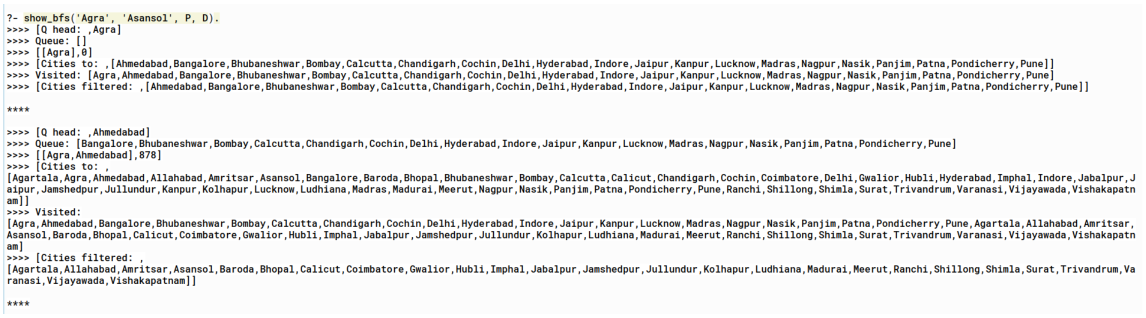


```
?- show_gbs('Agra', 'Asansol', P, D).
>>>> [PQ head: ,Agra]
>>>> PQueue: []
>>>> [Cities to: ,[Ahmedabad,Bangalore,Bhubaneshwar,Bombay,Calcutta,Chandigarh,Cochin,Delhi,Hyderabad,Indore,Jaipur,Kanpur,Lucknow,Madras,Nagpur,Nasik,Panjim,Patna,Pondicherry,Pune]]
>>>> Visited: [Agra,Ahmedabad,Bangalore,Bhubaneshwar,Bombay,Calcutta,Chandigarh,Cochin,Delhi,Hyderabad,Indore,Jaipur,Kanpur,Lucknow,Madras,Nagpur,Nasik,Panjim,Patna,Pondicherry,Pune]
>>>> [Cities filtered: ,[Ahmedabad,Bangalore,Bhubaneshwar,Bombay,Calcutta,Chandigarh,Cochin,Delhi,Hyderabad,Indore,Jaipur,Kanpur,Lucknow,Madras,Nagpur,Nasik,Panjim,Patna,Pondicherry,Pune]]

****

>>>> [PQ head: ,Calcutta]
>>>> PQueue: [Ahmedabad,Bangalore,Bhubaneshwar,Bombay,Chandigarh,Cochin,Delhi,Hyderabad,Indore,Jaipur,Kanpur,Lucknow,Madras,Nagpur,Nasik,Panjim,Patna,Pondicherry,Pune]
>>>> [Cities to: ,
[Agartala,Agra,Ahmedabad,Allahabad,Amritsar,Asansol,Bangalore,Baroda,Bhopal,Bhubaneshwar,Bombay,Calcutta,Calicut,Chandigarh,Cochin,Coimbatore,Delhi,Gwalior,Hubli,Hyderabad,Imphal,Indore,Jabalpur,J
aipur,Jamshedpur,Jullundur,Kanpur,Kolhapur,Lucknow,Ludhiana,Madras,Madurai,Meerut,Nagpur,Nasik,Panjim,Patna,Pondicherry,Pune,Ranchi,Shillong,Shimla,Surat,Trivandrum,Varanasi,Vijayawada,Vishakapatn
am]]
>>>> Visited:
[Agra,Ahmedabad,Bangalore,Bhubaneshwar,Bombay,Calcutta,Chandigarh,Cochin,Delhi,Hyderabad,Indore,Jaipur,Kanpur,Lucknow,Madras,Nagpur,Nasik,Panjim,Patna,Pondicherry,Pune,Agartala,Allahabad,Amritsar,
Asansol,Baroda,Bhopal,Calicut,Coimbatore,Gwalior,Hubli,Imphal,Jabalpur,Jamshedpur,Jullundur,Kolhapur,Ludhiana,Madurai,Meerut,Ranchi,Shillong,Shimla,Surat,Trivandrum,Varanasi,Vijayawada,Vishakapatn
am]
>>>> [Cities filtered: ,
[Agartala,Allahabad,Amritsar,Asansol,Baroda,Bhopal,Calicut,Coimbatore,Gwalior,Hubli,Imphal,Jabalpur,Jamshedpur,Jullundur,Kolhapur,Ludhiana,Madurai,Meerut,Ranchi,Shillong,Shimla,Surat,Trivandrum,Va
ranasi,Vijayawada,Vishakapatnam]]
```

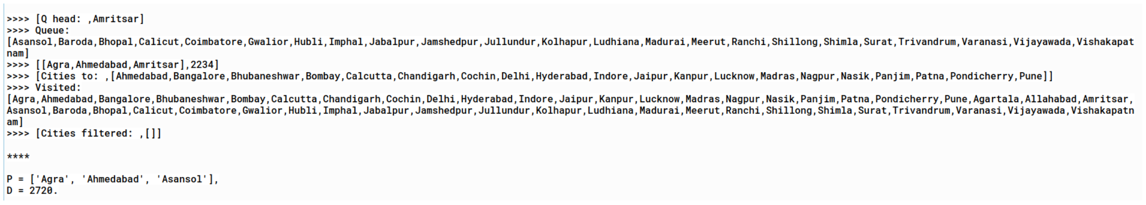


```
>>>> [PQ head: ,Baroda]
>>>> PQueue: [Agartala,Allahabad,Amritsar,Asansol]
>>>> [Cities to: ,[Ahmedabad,Bangalore,Bhubaneshwar,Bombay,Calcutta,Chandigarh,Cochin,Delhi,Hyderabad,Indore,Jaipur,Kanpur,Lucknow,Madras,Nagpur,Nasik,Panjim,Patna,Pondicherry,Pune]]
>>>> Visited:
[Agra,Ahmedabad,Bangalore,Bhubaneshwar,Bombay,Calcutta,Chandigarh,Cochin,Delhi,Hyderabad,Indore,Jaipur,Kanpur,Lucknow,Madras,Nagpur,Nasik,Panjim,Patna,Pondicherry,Pune,Agartala,Allahabad,Amritsar,
Asansol,Baroda,Bhopal,Calicut,Coimbatore,Gwalior,Hubli,Imphal,Jabalpur,Jamshedpur,Jullundur,Kolhapur,Ludhiana,Madurai,Meerut,Ranchi,Shillong,Shimla,Surat,Trivandrum,Varanasi,Vijayawada,Vishakapatn
am]
>>>> [Cities filtered: ,[]]

****

P = ['Agra', 'Calcutta', 'Asansol'],
D = 1526.
```

3. Breadth First Search



```
?- show_bfs('Agra', 'Asansol', P, D).
>>>> [Q head: ,Agra]
>>>> Queue: []
>>>> [[Agra],0]
>>>> [Cities to: ,[Ahmedabad,Bangalore,Bhubaneshwar,Bombay,Calcutta,Chandigarh,Cochin,Delhi,Hyderabad,Indore,Jaipur,Kanpur,Lucknow,Madras,Nagpur,Nasik,Panjim,Patna,Pondicherry,Pune]]
>>>> Visited: [Agra,Ahmedabad,Bangalore,Bhubaneshwar,Bombay,Calcutta,Chandigarh,Cochin,Delhi,Hyderabad,Indore,Jaipur,Kanpur,Lucknow,Madras,Nagpur,Nasik,Panjim,Patna,Pondicherry,Pune]
>>>> [Cities filtered: ,[Ahmedabad,Bangalore,Bhubaneshwar,Bombay,Calcutta,Chandigarh,Cochin,Delhi,Hyderabad,Indore,Jaipur,Kanpur,Lucknow,Madras,Nagpur,Nasik,Panjim,Patna,Pondicherry,Pune]]

****

>>>> [Q head: ,Ahmedabad]
>>>> Queue: [Bangalore,Bhubaneshwar,Bombay,Calcutta,Chandigarh,Cochin,Delhi,Hyderabad,Indore,Jaipur,Kanpur,Lucknow,Madras,Nagpur,Nasik,Panjim,Patna,Pondicherry,Pune]
>>>> [[Agra,Ahmedabad],878]
>>>> [Cities to: ,
[Agartala,Agra,Ahmedabad,Allahabad,Amritsar,Asansol,Bangalore,Baroda,Bhopal,Bhubaneshwar,Bombay,Calcutta,Calicut,Chandigarh,Cochin,Coimbatore,Delhi,Gwalior,Hubli,Hyderabad,Imphal,Indore,Jabalpur,J
aipur,Jamshedpur,Jullundur,Kanpur,Kolhapur,Lucknow,Ludhiana,Madras,Madurai,Meerut,Nagpur,Nasik,Panjim,Patna,Pondicherry,Pune,Ranchi,Shillong,Shimla,Surat,Trivandrum,Varanasi,Vijayawada,Vishakapatn
am]]
>>>> Visited:
[Agra,Ahmedabad,Bangalore,Bhubaneshwar,Bombay,Calcutta,Chandigarh,Cochin,Delhi,Hyderabad,Indore,Jaipur,Kanpur,Lucknow,Madras,Nagpur,Nasik,Panjim,Patna,Pondicherry,Pune,Agartala,Allahabad,Amritsar,
Asansol,Baroda,Bhopal,Calicut,Coimbatore,Gwalior,Hubli,Imphal,Jabalpur,Jamshedpur,Jullundur,Kolhapur,Ludhiana,Madurai,Meerut,Ranchi,Shillong,Shimla,Surat,Trivandrum,Varanasi,Vijayawada,Vishakapatn
am]
>>>> [Cities filtered: ,
[Agartala,Allahabad,Amritsar,Asansol,Baroda,Bhopal,Calicut,Coimbatore,Gwalior,Hubli,Imphal,Jabalpur,Jamshedpur,Jullundur,Kolhapur,Ludhiana,Madurai,Meerut,Ranchi,Shillong,Shimla,Surat,Trivandrum,Va
ranasi,Vijayawada,Vishakapatnam]]

****
```



```
>>>> [Q head: ,Amritsar]
>>>> Queue:
[Asansol,Baroda,Bhopal,Calicut,Coimbatore,Gwalior,Hubli,Imphal,Jabalpur,Jamshedpur,Jullundur,Kolhapur,Ludhiana,Madurai,Meerut,Ranchi,Shillong,Shimla,Surat,Trivandrum,Varanasi,Vijayawada,Vishakapat
nam]
>>>> [[Agra,Ahmedabad,Amritsar],2234]
>>>> [Cities to: ,[Ahmedabad,Bangalore,Bhubaneshwar,Bombay,Calcutta,Chandigarh,Cochin,Delhi,Hyderabad,Indore,Jaipur,Kanpur,Lucknow,Madras,Nagpur,Nasik,Panjim,Patna,Pondicherry,Pune]]
>>>> Visited:
[Agra,Ahmedabad,Bangalore,Bhubaneshwar,Bombay,Calcutta,Chandigarh,Cochin,Delhi,Hyderabad,Indore,Jaipur,Kanpur,Lucknow,Madras,Nagpur,Nasik,Panjim,Patna,Pondicherry,Pune,Agartala,Allahabad,Amritsar,
Asansol,Baroda,Bhopal,Calicut,Coimbatore,Gwalior,Hubli,Imphal,Jabalpur,Jamshedpur,Jullundur,Kolhapur,Ludhiana,Madurai,Meerut,Ranchi,Shillong,Shimla,Surat,Trivandrum,Varanasi,Vijayawada,Vishakapatn
am]
>>>> [Cities filtered: ,[]]

****

P = ['Agra', 'Ahmedabad', 'Asansol'],
D = 2720.
```

## Steps to Run

1. Open Prolog
2. Consult `main.pl`
3. Write the clause `form` to form the database
4. Search algos (Given Start & End, find Path & Dist).
    a. Depth First Search: `show_dfs(Start, End, Path, Dist)`
    b. Greedy Best First Search: `show_gbs(Start, End, Path, Dist)`
    c. Breadth First Search: `show_bfs(Start, End, Path, Dist)`