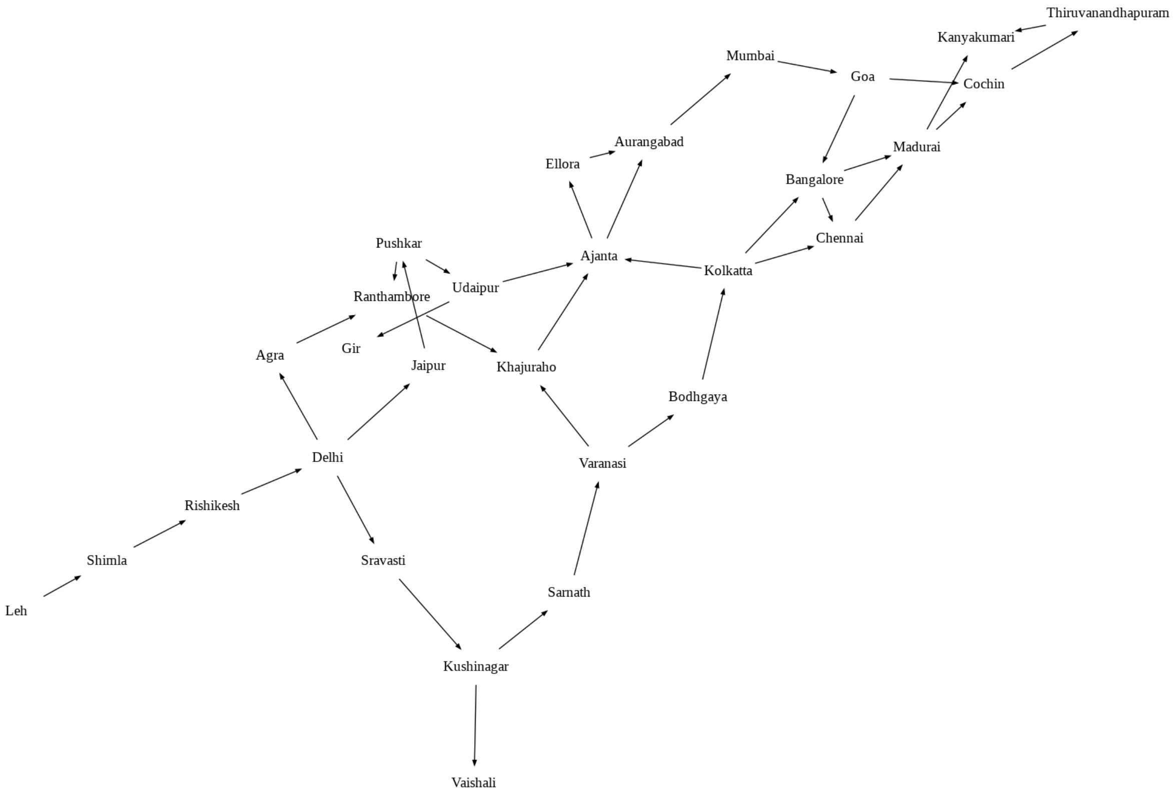
Week - 4, Graded Programming

Problem 3

Long journey

A tourist wants to travel around India from north to south. He has a policy that he never travels back towards the north. Write a Python function `longJourney(AList)` to find him a route with which he can visit the maximum number of cities according to his policy, where `AList` represents a graph of cities and routes between them. Every edge in adjacency list `AList` is a feasible route between one city to another from north to south. The function should return a list in the order the cities are to be visited to visit maximum cities.

An example of cities and route between them(as edge) is shown below.



Sample Adjacency List

```
{'Madurai': ['Cochin', 'Kanyakumari'],
 'Vaishali': [],
 'Varanasi': ['Khajuraho', 'Bodhgaya'],
 'Thiruvanandhapuram': ['Kanyakumari'],
 'Udaipur': ['Gir', 'Ajanta'],
 'Rishikesh': ['Delhi'],
 'Shimla': ['Rishikesh'],
 'Bangalore': ['Chennai', 'Madurai'],
 'Agra': ['Ranthambore'],
 'Ellora': ['Aurangabad'],
 'Bodhgaya': ['Kolkatta'],
 'Cochin': ['Thiruvanandhapuram'],
 'Pushkar': ['Udaipur', 'Ranthambore'],
 'Ranthambore': ['Khajuraho'],
 'Gir': [],
 'Aurangabad': ['Mumbai'],
 'Kolkatta': ['Ajanta', 'Bangalore', 'Chennai'],
 'Chennai': ['Madurai'],
 'Sravasti': ['Kushinagar'],
 'Leh': ['Shimla'],
 'Sarnath': ['Varanasi'],
 'Delhi': ['Jaipur', 'Agra', 'Sravasti'],
 'Goa': ['Cochin', 'Bangalore'],
 'Kanyakumari': [],
 'Kushinagar': ['Sarnath', 'Vaishali'],
 'Khajuraho': ['Ajanta'],
 'Jaipur': ['Pushkar'],
 'Mumbai': ['Goa'],
 'Ajanta': ['Ellora', 'Aurangabad']}
```

Sample Output

```
['Leh', 'Shimla', 'Rishikesh', 'Delhi', 'Sravasti', 'Kushinagar', 'Sarnath',
'Varanasi', 'Bodhgaya', 'Kolkatta', 'Ajanta', 'Ellora', 'Aurangabad',
'Mumbai', 'Goa', 'Bangalore', 'Chennai', 'Madurai', 'Cochin',
'Thiruvanandhapuram', 'Kanyakumari']
```