**2. In your Gandhinagar city, there are various locations such as Vidhan sabha, Achardham, Gandhi Asharam,...,etc. There exists a road network that connects all the locations. Due to elections in Gujarat and rallies, all the paths between any two locations act as one-way. You and your friends have decided to visit all the major locations as today is a holiday. You have to start your traveling plan from the hostel and after visiting each location you return to the hostel. The condition is you cannot follow the same path which you have already visited. Write an algorithm that gives you the efficient route to successfully execute your plan otherwise you drop your today’s plan.**

**Algorithm**

* Mark all the edges as unvisited and create and empty path and make the pathExist false.
* Start from the vertex v1 and visit the next vertex (use adjacency list).
* Keep track of visited edges.
* Add current vertex to result (taking integer array here) to keep track of path from vertex v1.
* Now if you look carefully, the new problem is to find paths from the current vertex to destination. For instance as per the example above, start from vertex 0 and visit vertex 1. Now all the paths from vertex 1 to vertex 5 will be included in our final result if we add vertex 0. So make a recursive call with source as vertex 1 and destination as vertex 5.
* Once reach to the destination vertex, print the path.
* Mark the current node as unmarked and delete it from path.
* Now visit the next node in adjacency list in step 1 and repeat all the steps (loop)

**Analysis**

This is a brute force approach, where each possibility is checked. Since, each location can be selected in any order. If we try finding the recursion tree, at, ith level there will be li recursive calls, where l is number of locations and since in each call DFS is called, therefore the overall complexity will be O(li. (V + E)), where V is the number of vertices and E is the number of edges.