Take Home Assignment 1

1. Algo implementation of :
   Display all paths in a directed acyclic graph and count the number of them.
   - Initialize an empty list to store paths
   - For v ϵ V {unexplored set of vertices} :

```
        Add the path traversed till now to list
        Recursively, move the the child of v,
        v=child(v), add child to path and path to list
```

Note : Since, it’s an acyclic graph, the recursive function should terminate
Time Complexity : O(V*E) , where E= no. of edges and V=no. of vertices

Efficient Approach : Memoization of paths starting from already visited nodes in a map has been implemented too. **Time complexity : O(V+E).**

- Initialize another empty list to store all the paths related to a single current node
- Add the list to a map and mark the node visited.
- Repeat the step 2 of the above algo but this time if (child(v) is visited) then
  Add all paths corresponding to the child(v) stored in map to v
- Add all in the global list.

2. Algo implementation of :
   Topological Sort -
   - Insert all vertices with indegree 0 in a Queue
   - While Queue is not empty

```
        U = Queue.Poll()
        Print(U)
        For v ϵ child(U)
        If indegree of v reducing by 1 is 0
        Add v to Queue
```

Time Complexity : same as BFS **- O(E+V)**, where E=no. of edges and V=no. of vertices

----

/*******************Abhik Chakraborty - 17114003*****************/