210.LeetCode

**Course Schedule II**

**Algorithm -Topological sort**

\*Same as 210.leetcode but we need to print the array

After running the for loop we pop from the stack and save it in the array and then we print it.

class Solution {

boolean isCyclic=false;

Stack<Integer> st=new Stack();

public int[] findOrder(int numCourses, int[][] prerequisites)

{

ArrayList<ArrayList<Integer>> graph=new ArrayList();

for(int i=0 ;i<numCourses; i++)

{

graph.add(new ArrayList());

}

for(int[] edge :prerequisites)

{

graph.get(edge[1]).add(edge[0]);

}

boolean[] visited=new boolean[numCourses];

boolean[] restack=new boolean[numCourses];

for(int i=0 ;i<numCourses ;i++)

{

dfs(i,visited,restack,graph);

}

if(isCyclic)

{

return new int[0];

}

int[] answer=new int[numCourses];

for(int i=0 ;i<numCourses ;i++)

{

answer[i]=st.pop();

}

return answer;

}

public void dfs(int i ,boolean[] visited ,boolean[] restack,ArrayList<ArrayList<Integer>> graph)

{

if(visited[i])

return;

if(restack[i])

isCyclic=true;

visited[i]=true;

restack[i]=true;

for(int x :graph.get(i))

{

if(!visited[x])

{

dfs(x,visited,restack,graph);

}

}

restack[i]=false;

st.push(i);

}

}