**Topological Sort**

#include <bits/stdc++.h>

void dfsTopo(unordered\_map<int,list<int>>&adj,vector<bool>&visited,stack<int>&s,int node){

visited[node] = 1;

for(auto neigh: adj[node]){

if(!visited[neigh]){

dfsTopo(adj,visited,s,neigh);

}

}

s.push(node);

}

vector<int> topologicalSort(vector<vector<int>> &edges, int v, int e) {

// preparing a adjacency list

unordered\_map<int,list<int>>adj;

for(int i=0;i<edges.size();i++){

int u = edges[i][0];

int v = edges[i][1];

adj[u].push\_back(v);

}

vector<bool>visited(v);

stack<int>s;

for(int i=0;i<v;i++){

if(!visited[i]){

dfsTopo(adj,visited,s,i);

}

}

vector<int>ans;

while(!s.empty()){

ans.push\_back(s.top());

s.pop();

}

return ans;

}