Topological Sort : Kahn's Algorithm

```
using namespace std;
using ll = long long;

int nodes, edges;
vector< vector<int> >adj;
vector<int> indegree;

void topologicalSortKhan() {
    queue<int> q;
    for (int i = 0; i < nodes; ++i) if (indegree[i] == 0) q.push(i);
    int cnt = 0;
    vector<int> order;
    while (!q.empty()) {
        int u = q.front(); q.pop();
        ++cnt;
        order.push_back(u);
        for (int i = 0; i < (int) adj[u].size(); ++i) {
            int v = adj[u][i];
            --indegree[v];
            if (indegree[v] == 0) q.push(v);
        }
    }
    if (cnt != nodes) {
        cout << "There exists a cycle in the graph!" << endl;
    } else {
        for (auto x : order) cout << x << " ";
        cout << endl;
    }
}

int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(NULL); cout.tie(NULL);

    cin >> nodes >> edges;
    adj.resize(nodes);
    indegree.resize(nodes);

    int u, v;
    for (int i = 0; i < edges; ++i) {
        cin >> u >> v;
        adj[u].push_back(v);
        ++indegree[v];
    }

    topologicalSortKhan();


    return 0;
}
```

Topological Sort - Finding all possible paths

- Editorial > tag: backtracking

```
#include <bits/stdc++.h>
#include <stdio.h>
#define fin freopen("input", "r", stdin)
#define endl "\n"
```

```
#define whatis(x) cerr << #x << ": " << x << endl;

using namespace std;
using ll = long long;

int nodes, edges;
vector< vector<int> > adj;
vector<int> indegree;

void topologicalSort(vector<int> &res, vector<bool> &vis) {
    bool flag = false;
    for (int i = 0; i < nodes; ++i) {
        if (indegree[i] == 0 && !vis[i]) {
            for (int j = 0; j < (int) adj[i].size(); ++j) {
                --indegree[adj[i][j]];
            }
            res.push_back(i);
            vis[i] = true;
            topologicalSort(res, vis);
            // resetting visited, res and indegree for
            // backtracking
            vis[i] = false;
            res.erase(res.end()-1);
            for (int j = 0; j < (int) adj[i].size(); ++j) {
                ++indegree[adj[i][j]];
            }
            flag = true;
        }
    }
    if (!flag)
    {
        for (int i = 0; i < (int) res.size(); i++)
            cout << res[i] << " ";
        cout << endl;
    }
}

void topologicalSortUtil() {
    vector<bool> vis(nodes);
    for (int i = 0; i < nodes; ++i) vis[i] = false;
    vector<int> res;
    topologicalSort(res, vis);
}

int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(NULL); cout.tie(NULL);

    int u, v;
    cin >> nodes >> edges;
    adj.resize(nodes);
    indegree.resize(nodes);
    for (int i = 0; i < edges; ++i) {
        cin >> u >> v;
        adj[u].push_back(v);
        ++indegree[v];
    }
    topologicalSortUtil();
}
```