Topological Sorting using Kahn's Algorithm

1 Introduction

Topological Sorting is a linear ordering of vertices in a Directed Acyclic Graph (DAG) such that for every directed edge $u \to v$, vertex u appears before v in the ordering. It is widely used in dependency resolution, task scheduling, and analyzing course prerequisites.

2 Algorithm: Kahn's Algorithm (BFS)

Kahn's Algorithm is an efficient way to perform topological sorting using a queue. The main steps are:

1. Compute the in-degree of each vertex.
2. Enqueue all vertices with in-degree 0.
3. Process the queue:
   (a) Dequeue a vertex and add it to the topological order.
   (b) Reduce the in-degree of all its adjacent vertices.
   (c) If any vertex's in-degree becomes 0, enqueue it.
4. If all vertices are processed, the graph is a DAG; otherwise, it contains a cycle.

3 C++ Implementation

```
#include <bits/stdc++.h>
using namespace std;

class Solution {
public:
    vector<int> topo(int N, vector<int> adj[]) {
        queue<int> q;
        vector<int> indegree(N, 0);
        for(int i = 0; i < N; i++) {
```

```
            for(auto it: adj[i]) {
                indegree[it]++;
            }
        }

        for(int i = 0; i < N; i++) {
            if(indegree[i] == 0) {
                q.push(i);
            }
        }

        vector<int> topo;
        while(!q.empty()) {
            int node = q.front();
            q.pop();
            topo.push_back(node);
            for(auto it : adj[node]) {
                indegree[it]--;
                if(indegree[it] == 0) {
                    q.push(it);
                }
            }
        }
        return topo;
    }
};

int main() {
    vector<int> adj[6];
    adj[5].push_back(2);
    adj[5].push_back(0);
    adj[4].push_back(0);
    adj[4].push_back(1);
    adj[3].push_back(1);
    adj[2].push_back(3);

    Solution obj;
    vector<int> v = obj.topo(6, adj);
    for(auto it : v)
        cout << it << "␣";
    return 0;
}
```

4 Complexity Analysis

- Calculating in-degree: $O(V + E)$
- Processing queue: $O(V + E)$
- Overall complexity: $O(V + E)$

5 Applications

- Task Scheduling
- Course Prerequisite Resolution
- Dependency Resolution in Package Managers
- Circuit Design