ASSIGNMENT 4

Topological Sort Using In-Degree (Kahn's Algorithm)

CSA0303 – DATA STRUCTURES FOR Problem Solving

NAME: KAVYA SHRI G

REG.NO: 192421052

AIM: To perform topological sorting on a Directed Acyclic Graph (DAG) using in-degree and queue-based Kahn's algorithm.

ALGORITHM:

1. Initialize the adjacency matrix for the DAG.
2. Calculate the in-degree for each node (number of incoming edges).
3. Enqueue all nodes with in-degree 0.
4. While the queue is not empty, dequeue a node and add it to the topological order.
5. Decrease the in-degree of all adjacent nodes and enqueue those that become 0.
6. Repeat until all nodes are processed and output the topological order

CODE:

```
#include <stdio.h>

#include <stdlib.h>

#define MAX 10

int main()

{

   int n = 6;

   int graph[MAX][MAX] = {0};

   int in_degree[MAX] = {0};
```

```
int queue[MAX], front = 0, rear = 0;
graph[1][2] = 1;
graph[1][3] = 1;
graph[3][4] = 1;
graph[2][4] = 1;
graph[4][5] = 1;
graph[5][6] = 1;
for (int i = 1; i <= n; i++)
   for (int j = 1; j <= n; j++)
      if (graph[i][j])
         in_degree[j]++;
for (int i = 1; i <= n; i++)
   if (in_degree[i] == 0)
      queue[rear++] = i;
printf("Topological Order: ");
while (front < rear) {
   int node = queue[front++];
   printf("%d ", node);
   for (int j = 1; j <= n; j++)
   {
      if (graph[node][j])
      {
         in_degree[j]--;
         if (in_degree[j] == 0)
            queue[rear++] = j;
      }
```

```
        }
    }
    printf("\n");
    return 0;
}
```

OUTPUT