Lab Program 4.

Topological Sort using DFS:

```
void dfs (int v) {
    visited [v] = 1;
    for (int i=0; i<n; i++) {
        if (adj[v][i] && !visited [i]) {
            dfs(i);
        }
    }
    push(v);
}

void topologicalSort() {
    for (int i=0; i<n; i++)
        visited [i] = 0;

    for (int i=0; i<n; i++) {
        if (!visited [i])
            dfs (i);
    }
    while (top != -1) {
        printf("%d", pop());
    }
    printf ("\n");
}
```

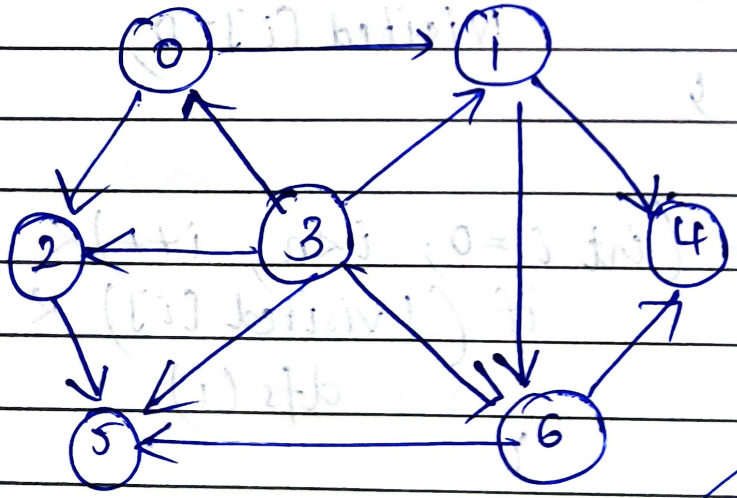
Date __/__/__
Page

Output:

Enter the number of nodes : 7
Enter the adjacency matrix :

0 1 1 0 0 0 0
0 0 0 0 1 0 1
0 0 0 0 0 1 0
1 1 1 0 0 1 1
0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 1 1 0

Topological order of node : 3 0 2 1 6 5 4



Source Removal Method

```
void degree (int adj [][20], int n) {
    int indegree [20];
    int sum=0;
    for (int j = 0; j<n ; j++) {
        sum=0;
        for(int i=0 ; i<n ; i++){
            sum= sum+ adj [i][j];
        }
        indegree [j] = sum;
    }
for (int i=0 ; i<n ; i++) {
    if (indegree [i] == 0) {
        top ++;
        st[top] = i;
    }
}

while (top != -1) {
    int u = st[top];
    top--;
    printf ("%d", u);
    for(int v=0; v<n; v++){
        if(adj [u][v] == 1) {
            indegree [v]--;
            if (indegree [v]==0) {
                top++;
                st[top] = v;
            }
        }
    }
}
```

Output:

Enter the number of nodes: 4

Enter the adjacency matrix:

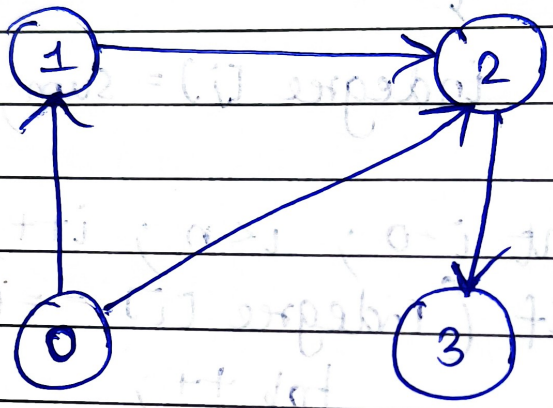
0 1 1 0

0 0 1 0

0 0 0 1

0 0 0 0

Topological order of nodes: 0 1 2 3



23/5/24