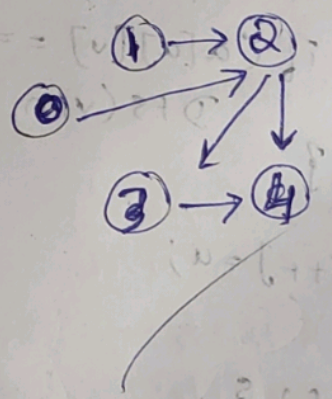
DFS - Topological order ; LAB - (4) 23/5/24 :

```
#include <stdio.h>
#include <stdlib.h>
#define MAX_VERTICES 100
int s[MAX_VERTICES] = {0};
int res[MAX_VERTICES];
int j = 0;

void DFS(int u, int n, int a[MAX_VERTICES][MAX_VERTICES])
{
    s[u] = 1;
    for(int v = 0 ; v < n; v++){
        if (a[u][v] == 1 && s[v] == 0) {
            DFS(v, n, a);
        }
    }
    res[j++] = u;
}

int main() {
    int n;
    printf("Enter the number of vertices: ");
    scanf("%d", &n);

    int a[MAX_VERTICES][MAX_VERTICES];
    printf("Enter the adjacency matrix:\n");
    for(int i = 0; i < n ; i++){
        for (int j = 0; j < n ; j++){
            scanf("%d", &a[i][j]);
        }
    }
for(int u = 0; u < n ; u++){
    if (s[u] == 0){
        DFS(u, n, a);
    }
}
```

```
    printf("Topological order : ");
    for(int i=j-1 ; i>=0 ; i--){
        printf("%d", res[i]);
    }
    printf("\n");
    return 0;
}
```

Output:

```
Enter the number of vertices :5
Enter the adjacency matrix:
0 0 1 0 0
0 0 1 0 0
0 0 0 1 1
0 0 0 0 1
0 0 0 0 0
Topological order : 0 2 3 4
```



Source Removal method - Topological ~~ord~~ sort:

```
#include <stdio.h>
#include <stdlib.h>

int st[100];
int top = -1;
void degree(int adj[][20], int n){
    int indegree[20];
    int sum=0;
    for(int j=0; j<n; j++){
        sum=0;
```

```
        for (int j = 0 ; j<n ; j++) {
            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;
                }
            }
        }
    }
}
int main() {
    int n;
    printf("Enter the number of nodes: ");
    scanf("%d", &n);
    int adj[20][20];
```

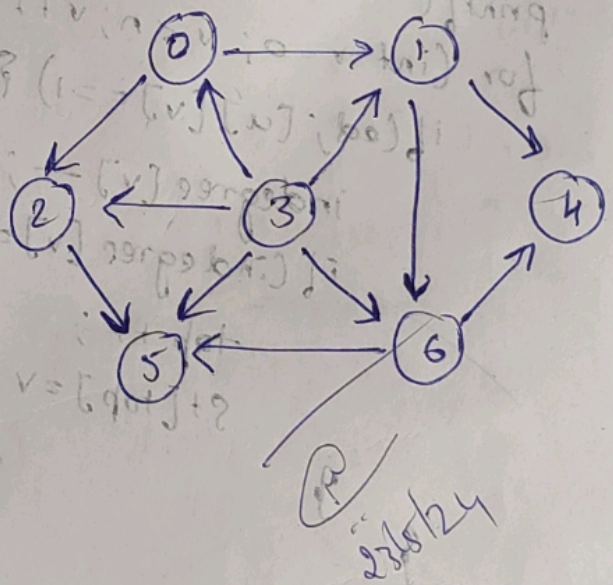
```
    printf("Enter the adjacency matrix:\n");
    for (int i=0; i<n; i++) {
        for (int j=0; j<n; j++) {
            scanf("%d", &adj[i][j]);
        }
    }
    printf("Topological order of nodes:");
    degree(adj, n);
    return 0;
}
```

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 nodes: 3021654