23/05/24 Week-4

Topological Sort using source Removal method.

```
#include<stdio.h>
#include<stdlib.h>
#define MAX_VERTICES 10.
void topological_sort(int a[MAX_VERTICES][MAX_VERTICES],
                      int n){
    int indegree[MAX_VERTICES]={0};
    int S[MAX_VERTICES];
    int top=-1;
    for(int i=0; i<n; i++){
        int sum=0;
        for(int j=0; j<n; j++){
            sum+=a[j][i];
        }
        indegree[i]=sum;
    }
    for(int i=0; i<n; i++){
        if(indegree[i]==0){
            top++;
            S[top]=i;
        }
    }
    int T[MAX_VERTICES];
    int t_index=0;
    while(top!=-1){
        int u=S[top];
        top--;
        T[t_index++]=u;
        for(int v=0; v<n; v++){
            if(a[u][v]==1){
                indegree[v]--;
```

```
        if(indegree[v]==0){
            top++;
            s[top]=v;
        }
    }
}
printf("Topological sequence:");
for(int i=0; i<t_index; i++){
    printf("%d", T[i]);
}
printf("\n");
}
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]);
    }
}
topological_sort(a,n);
return 0;
}
```

Output:

Enter the number of vertices: 5

Enter adjacency matrix:

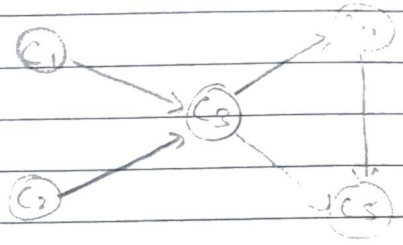
0 0 1 0 0

0 0 1 0 0

0 0 0 1 1

0 0 0 0 1

0 0 0 0 0.



Topological sequence: 1 0 2 3 4.

C2 C1 C3 C4 C5

→ Topological order using DFS.

```
#include <stdio.h>
#include <stdlib.h>
#define MAX_VERTICES 10.
int S[MAX_VERTICES]={0};
int res[MAX_VERTICES]0;
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;
}
```

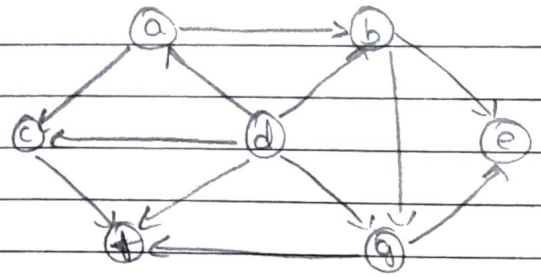
23/5/24

Output:

Enter the number of vertices: 7

Enter 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: 3 0 2 1 6 5 4

d a c b g e f