

## Topological sort (Source removal method)

```
#include <stdio.h>
#define v 100
int top=-1;
void indegree(int a_matrix[v][v],int n,int in[v])
{
    for(int i=0;i<n;i++)
    {
        for(int j=0;j<n;j++)
        {
            if(a_matrix[i][j])
            {
                in[j]++;
            }
        }
    }
}
void toposort(int a_matrix[v][v],int n)
{
    int in[v]={0};
    int topo[v];
    int k=0;

    int s[v]={0};
    indegree(a_matrix,n,in);
    for(int i=0;i<n;i++)
    {
        if(in[i]==0)
        {
            top++;
            s[top]=i;
        }
    }
    while(top!=-1)
    {
        int vertex=s[top];
        top--;
        topo[k++]=vertex;
        for(int i=0;i<n;i++)
        {
            if(a_matrix[vertex][i])
            {
                in[i]--;
                if(in[i]==0)
            }
        }
    }
}
```

```

        {
            top++;
            s[top]=i;
        }
    }
}
}
if(k!=n)
{
    printf("cycle exists");
}
else{
    printf("the topological sort:");
    for(int i=0;i<n;i++)
    {
        printf("%d ",topo[i]+1);
    }
}
}
int main()
{
    int a_matrix[v][v];
    int n;
    printf("enter the no of vertices:");
    scanf("%d",&n);
    printf("enter the adjaceny matrix:\n");
    for(int i=0;i<n;i++)
    {
        for(int j=0;j<n;j++)
        {
            scanf("%d",&a_matrix[i][j]);
        }
    }
    toposort(a_matrix,n);
    return 0;
}
Output:

```

```

enter the no of vertices:5
enter the adjaceny 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
the topological sort:2 1 3 4 5

```

## Topological Sort(DFS)

```
#include <stdio.h>
#define v 100
int j=0;
void dfs(int a_matrix[v][v],int n,int visited[],int start,int res[])
{
    visited[start]=1;
    for(int i=0;i<n;i++)
    {
        if(a_matrix[start][i]==1&& visited[i]==0 )
        {
            dfs(a_matrix,n,visited,i,res);
        }
    }

    res[j++]=start;
}

void toposort(int a_matrix,int n)
{
    int visited[v]={0};
    int res[v];
    j=0;
    for(int i=0;i<n;i++)
    {
        if(visited[i]==0)
        {
            dfs(a_matrix,n,visited,i,res);
        }
    }
    printf("the topological sort:");
    for(int i=n-1;i>=0;i--)
    {
        printf("%d",res[i]);
    }
}

int main()
{
    int a_matrix[v][v];
    int n;

    printf("enter the no of vertices:");
    scanf("%d",&n);
    printf("enter the adjaceny matrix:\n");
    for(int i=0;i<n;i++)
    {
```

```
        for(int j=0;j<n;j++)
        {
            scanf("%d",&a_matrix[i][j]);
        }
    }

    toposort(a_matrix,n);
    return 0;
}
```

### Output:

```
enter the no of vertices:4
enter the adjacency matrix:
0 0 0 0
1 0 0 0
1 0 0 1
0 1 0 0
the topological sort:2310
```