23/05/24

## Lab-4

Topological sort algorithm using Source Removal method

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

void topologicalSort (int **a, int n) {
int indegree [n]

for (int j =0 ; j <n; j++) {
   int sum =0;
  for (int i =0; i<n; i++) {
  sum + = a [i][j] ;
   }
  indegree [j] = sum;
   }
 for (int i= 0; i <n ; i++) {
     if (indegree [i] == 0) {
       s [++ top ] = i;

   }
  }

While (top ! = -1){
int v = s[top --];
   T [k++ ] = u;

for (int v =0; v < n ; v++) {
  if (a[u] [v ] = =1 ) {
if (indegree [v] == 0) {
        s[++top] = v;
```

```
    }
  }
 }
}
Printf ("Topological Order :");
 for (int i=0; i< k; i++) {
 Printf ("%d", T[i]);
  }
 Printf ("\n");
 }
 int main() {
  int n;
 Printf ("Enter the number of Vertices:");
 Scanf ("%d", &n);
 int ** a = (int **) malloc (n * sizeof (int *));
 for (int i=0; i<n; i++) {
 a[i] = (int *) malloc (n * sizeof (int));
 }
 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 Store (a, n);
 for (int i = 0; i<n; i++) {
    free (a[i]);
   }
   free (a);
   return 0;
  }
```

output:

Enter the number of vertices 6

Enter the adjacency matrix:

0 0 1 1 0 0

0 0 0 1 1 0

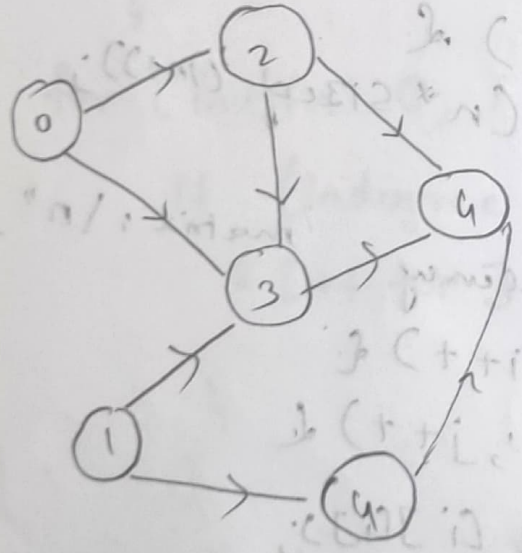
0 0 0 1 0 1

0 0 0 0 0 1

0 0 0 0 0 1

0 0 0 0 0 0

Topological Order: 1 4 0 2 3 5




| | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 0 | |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 2 | 0 | 0 | 0 | 1 | 0 | 1 |
| 3 | 0 | 0 | 0 | 0 | 0 | 1 |
| 4 | 0 | 0 | 0 | 0 | 0 | 1 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0 | 0 | 1 | 3 | 1 | 3 |

2) Topological Sort Algorithm using DFS..

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

void DFS (int u, int n, int **a, int *s,
          int *res, int *j) {

    s[u] = 1;
    for (int v = 0; v<n; v++) {
    if (a[u][v] == 1 && s[v] == 0) {
      DFS (v, n, a, s, res, j);
    }
    }
    res[(*j)++] = u;
}
void topologicalOrder (int n, int **a) {
    int s[n];
    int res[n];
    int j = 0;
    for (int i = 0; i<n; i++) {
      if (s[u] == 0) {
      DFS (u, n, a, s, res, &j);
      }
    }
    printf ("Topological Order : ");
    for (int i = n-1; i >= 0; i--) {
      printf ("%d ", res[i]);
    }
```

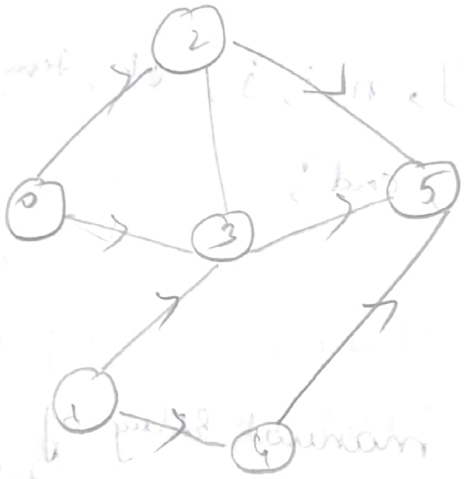
```
printf ("\n");
}

int main() {
    int n;
    printf ("Enter the number of vertices:");
    scanf ("%d", &n);
int ** a = (int **) malloc (n * sizeof (int *));
for (int i = 0; i < n; i++) {
for (int j = 0; j < n; j++) {
scanf ("%d", &a[i][j]);
}
}
topological Order (n, a)
    for (int i = 0; i < n; i++) {
    free (a[i]);
    }
    free (a);
    return 0;
}
```

Outputs:

```
Enter the number of vertices 6
Enter the adjacency matrix:
```

| | | | | | |
|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 |

Topological order: 1 4 0 2 3 5




P
23/5/24