20/5/21

Page No: 99
Date:

Q → Topological Sorting

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
int a[10][10], indegree[10], front = -1, rear = -1,
stack[10], t[10], k, n;
void get(int n) {
    int i,j;
    printf("Enter adjacency matrix : \n");
    for(i = 0; i < n; i++) {
        printf("Enter row %d \n", (i+1));
        for(j = 0; j < n; j++)
            scanf("%d", &a[i][j]);
    }
}
void cal_indegree(int n) {
    int i, j;
    for(i = 0; i < n; i++) {
        for(j = 0; j < n; j++) {
            indegree[i] = indegree[i] + a[j][i];
}
void push(int x) {
    if(front == -1 && rear == -1)
        front = rear = 0;
    else if(rear == n-1) { return; }
```

```
    else { rear++; }
    stack[rear] = x;
}
int pop() {
    int x;
    if(front == -1 || front > rear) { return -1; }
    x = stack[front];
    if(front == rear || front > rear) {
        front = rear = -1; }
    else { front++; }
    return x;
}
int main() {
    int i, n;
    printf("Enter no. of vertices : ");
    scanf("%d", &n);
    get(n);
    for(i=0; i<n; i++) {
        indegree[i] = 0; }
    double t = time = 0.0;
    clock_t begin = clock();
    cal_indegree(n);
    for(int i=0; i<n; i++)
        if(indegree[i] == 0) {
            push(i);
```

```
while (front != -1) {
    int u = pop();
    if (u == -1) { pop(t) break; }
    t[k] = u;
    k++;
    for (v = 0; v < n; v++) {
        if (a[v][u] == 1) {
            ((indegree[v])--);
            if (indegree[v] == 0) {
                push(v); }
        }
    }
}
clock_t end = clock();
t_time += (double)(end - begin) / (clocks_PER_SEC);
printf("Order :");
for (i = 0; i < k; i++) {
    printf("%d\t", t[i]); }
printf("\n for matrix on = %d \t
    Time = %f, n, t_time);
return 0;
}
```