Third Year B. Tech., Sem V 2022-23

Design and Analysis of Algorithm Lab

Assignment / Journal submission

PRN/ Roll No: 21520010

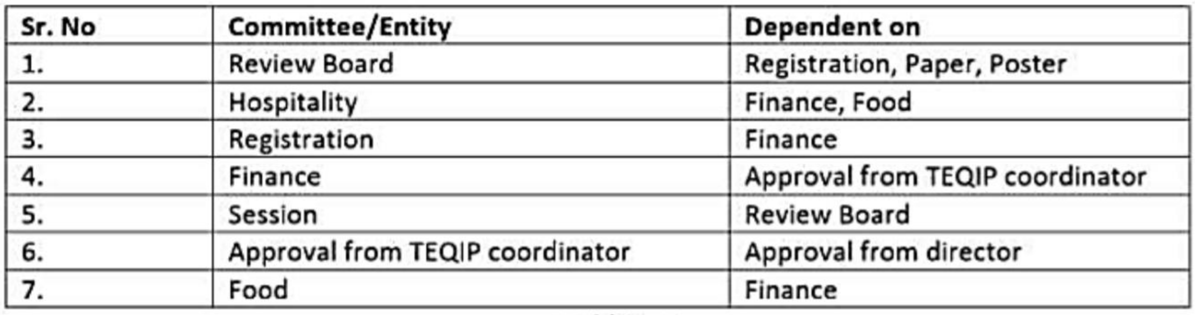
Full name: Mayur Sunil Savale

Batch: T8

Assignment: Week 11

Title of assignment: Graph Algorithm

1. One day ACSES president decides to organize Research Symposium in WCE. He proposed his idea to all ACSES members, and all agreed on his idea.
   Secretary of ACSES has ready bird view plan which contains all Symposium committees in the form of dependencies, with which proper execution of plan can be done. Table shows committees/ entities and their dependencies.



Sr. No	Committee/Entity	Dependent on
1.	Review Board	Registration, Paper, Poster
2.	Hospitality	Finance, Food
3.	Registration	Finance
4.	Finance	Approval from TEQIP coordinator
5.	Session	Review Board
6.	Approval from TEQIP coordinator	Approval from director
7.	Food	Finance

   Now president has divided members of ACSES into these committees/entities and he wants to tell them in which order to execute each committees/entity’s task for smooth execution.

Implement an algorithm to help ACSES president to design all possible linear order execution of committee/entity. Convert this information in the form of Dependence Graph and apply Topological sort with Discovering and Finishing time.

Ans:

a) **Algorithm: (Pseudocode)**

- ➢ Compute in-degree (number of incoming edges) for each of the vertex present in the DAG and initialize the count of visited nodes as 0.
- ➢ Pick all the vertices with in-degree as 0 and add them into a queue (Enqueue operation).
- ➢ Remove a vertex from the queue (Dequeue operation) and then.
    A. Increment count of visited nodes by 1.
    B. Decrease in-degree by 1 for all its neighbouring nodes.
    C. If in-degree of a neighbouring nodes is reduced to zero, then add it to the queue.
- ➢ Repeat Step 3 until the queue is empty.
- ➢ If count of visited nodes is not equal to the number of nodes in the graph then the topological sort is not possible for the given graph.

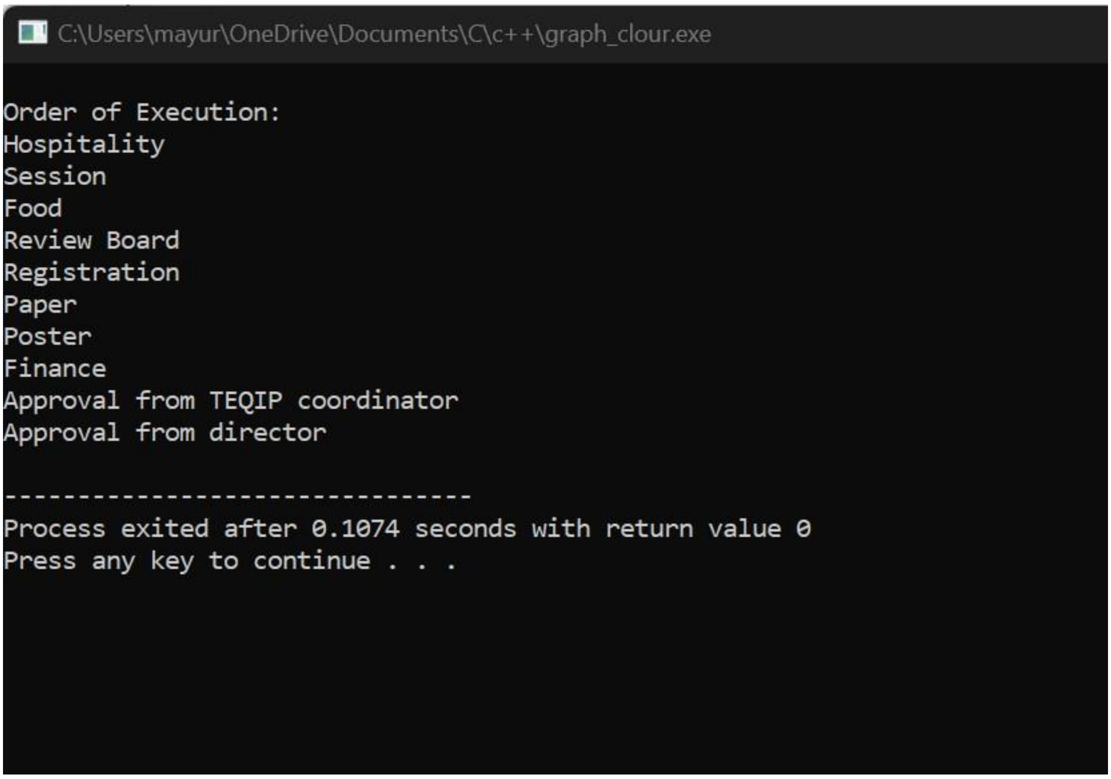
b) **Code snapshots of implementation**

```
#include <bits/stdc++.h>
using namespace std;

void topologicalsort(bool graph[10][10],vector<int>&ans)
{
   int in[10];
   for(int i=0;i<10;i++)
      in[i]=0;
   for(int i=0;i<10;i++)
   {
      for(int j=0;j<10;j++)
      {
           if(graph[i][j])
               in[j]++;
      }
   }
   queue<int>q;
   for(int i=0;i<10;i++)
   {
      if(in[i]==0)
           q.push(i);
   }
   while(!q.empty())
   {
      int p=q.front();
      q.pop();
      ans.push_back(p);
      for(int j=0;j<10;j++)
      {
           if(graph[p][j]==1)
           {
               in[j]--;
               if(in[j]==0)
                   q.push(j);
           }
      }
   }
}

int main()
{
   string p[10]={"Review Board", "Hospitality", "Registration", "Finance",
"Session", "Approval from TEQIP coordinator", "Food", "Paper",
"Poster", "Approval from director"};
   bool graph[10][10]={{0,0,1,0,0,0,0,1,1,0},
                       {0,0,0,1,0,0,1,0,0,0},
                       {0,0,0,1,0,0,0,0,0,0},
                       {0,0,0,0,0,1,0,0,0,0},
                       {1,0,0,0,0,0,0,0,0,0},
                       {0,0,0,0,0,0,0,0,0,1},
                       {0,0,0,1,0,0,0,0,0,0},
                       {0,0,0,0,0,0,0,0,0,0},
                       {0,0,0,0,0,0,0,0,0,0},
                       {0,0,0,0,0,0,0,0,0,0}};
   vector<int>ans;
   topologicalsort(graph,ans);
   cout << "\nOrder of Execution:\n";
   for(int i=0;i<ans.size();i++)
      cout << p[ans[i]] << "\n";
}
```

Output:



```

Order of Execution:
Hospitality
Session
Food
Review Board
Registration
Paper
Poster
Finance
Approval from TEQIP coordinator
Approval from director

--------------------------------
Process exited after 0.1074 seconds with return value 0
Press any key to continue . . .
```

c) Complexity of proposed algorithm (Time & Space)

- Time Complexity: O(V+E)
- Space Complexity: O(V)

d) Your comment (How your solution is optimal?)

- The outer for loop will be executed V number of times and the inner for loop will be executed E number of times, Thus overall time complexity is O(V+E). Topological Sorting for a graph is not possible if the graph is not a DAG.