# Department of Computing

# School of Electrical Engineering and Computer Science

**CS-250: Data Structure and Algorithms**

**Class: BSCS 10C**

**Lab 12:  Topological Sort**

**Date: 17th December, 2021**

**Time: 9:00 am – 11:50 am**

# Instructor: Prof. Dr. Faisal Shafait

# Lab Engineer: Mr. Aftab Farooq

# Lab 12: Topological Sort

**Introduction**

This lab is based on the implementation of Topological Sort.

**Objectives**

The objective of this lab is to become familiar with implementation of Topological Sort.

**Tools/Software Requirement**

Visual Studio C++

**Description**

A **topological sorting** for Directed Acyclic Graph (DAG) is a linear ordering of vertices such that for every directed edge uv, vertex u comes before v in the ordering. Topological Sorting for a graph is not possible if the graph is not a DAG.

**Lab Tasks**

****

**Task 1**

Write a C++ program to print topological sorting of a Directed Acyclic Graph (DAG). Consider a graph given in the above diagram

* Implement a function for adding an edges in a graph
* After creating a graph write a function that do topological sort
* In topological sort method Mark all the vertices as not visited
* Create and Call the recursive helper function to store Topological so sort starting from all vertices one by one
* In recursive helper function Mark the current node as visited.
* Push current vertex to stack which stores result
* At end prints a Topological Sort of the complete graph

**Task 2**Take two more differentDAG Graphs and print their topological sort by using same above code.

The two task are done by one code as the code is below.

**Important Note:** Practice your knowledge of OOP with C++ when creating a solution.

**Lab Grading:**

|  |  |
| --- | --- |
| **Task** | **Marks** |
| Lab Viva/Quiz | 5 |
| Comments/ Indentation | 2 |
| Solution Document | 2 |
| Output Screen Shots | 1 |
| -- | -- |
| Total | 10 |

**Deliverables**

This lab grading policy is as follows: The lab is graded between 0 to 10 marks. The submitted solution can get a maximum of 5 marks. At the end of each lab or in the next lab, there will be a viva related to the tasks. The viva has a weightage of 5 marks. Insert the solution/answer in this document. You must show the implementation of the tasks in the designing tool, along with your complete Word document to get your work graded. You must also submit this Word document on the LMS. In case of any problems discuss it by emailing it to [aftab.farooq@seecs.edu.pk](mailto:aftab.farooq@seecs.edu.pk).

**Note:** Students are required to upload the lab on LMS before deadline.

**Code:**

#include<iostream>

#include<conio.h>

#include<vector>

#include<stack>

using namespace std;

stack<int> s;

class Node{

public :

  vector<int> parent;

  int vertex;

  Node \*next;

};

class List{

public :

  void insert(int p, Node\* head, Node\* child){

    Node \*h = new Node();

    h = head;

    while (h->next){

      h = h->next;

    }

    Node\* inserted = new Node();

    child->parent.push\_back(p);

    inserted->parent.push\_back(p);

    inserted->vertex = child->vertex;

    inserted->next = NULL;

    h->next = inserted;

  }

  void display(Node\* head){

    Node\* temp = new Node();

    temp = head;

    while (temp){

      cout << temp->vertex << "---->";

      temp = temp->next;

    }

  }

  void topological\_sort(vector<Node\*> heads){

    if (heads.size() == 0){

      return;

    }

    cout << heads.size() << endl;

    int i, j, x, y, a;

    for (i = 0; i < heads.size(); i++){

      if (heads.at(i)->parent.size() == 0){

        s.push(heads.at(i)->vertex);

        for (a = 0; a < heads.size(); a++){

          Node \* node = new Node();

          node = heads.at(a);

          while (node){

            for (j = 0; j < node->parent.size(); j++){

              if (node->parent.at(j) == heads.at(i)->vertex){

                node->parent.erase(node->parent.begin() + j);

              }

            }

            node = node->next;

          }

        }

        heads.erase(heads.begin() + i);

        i--;

        for (x = 0; x < heads.size(); x++){

          for (y = 0; y < heads.at(x)->parent.size(); y++){

            cout << heads.at(x)->parent.at(y) << ",";

          }

          cout << " --> " << heads.at(x)->vertex;

          cout << endl;

        }

        cout << endl;

        cout << heads.size() << endl;

      }

    }

    topological\_sort(heads);

  }

};

int main(){

  vector<Node\*> heads;

  int i;

  for (i = 0; i <= 5; i++){

    Node\* temp = new Node();

    temp->vertex = i;

    temp->next = NULL;

    heads.push\_back(temp);

  }

  List \*list = new List();

  list->insert(5, heads.at(5), heads.at(2));

  list->insert(5, heads.at(5), heads.at(0));

  list->insert(4,heads.at(4), heads.at(0));

  list->insert(4,heads.at(4), heads.at(1));

  list->insert(2,heads.at(2), heads.at(3));

  list->insert(3,heads.at(3), heads.at(1));

  int j;

  list->topological\_sort(heads);

  cout << heads.size() << endl;

  cout << endl;

  cout << endl;

  while (!s.empty()){

    cout << s.top() << endl;

    s.pop();

  }

  \_getch();

  return 0;

}

**Output:**

**Text

Description automatically generated with low confidence**

Use proper indentation and comments. Lack of comments and indentation will result in deduction of marks.