Experiment - 6

Student Name: Pankaj Singh Kanyal
UID: 20BCS6668
Branch: AIML
Section/Group: AIML 4 B
Semester: 5th
Date of Performance: 11/10/2022
Subject Name: Advanced Programming Lab
Subject Code: 20CSP-334

1. AIM:

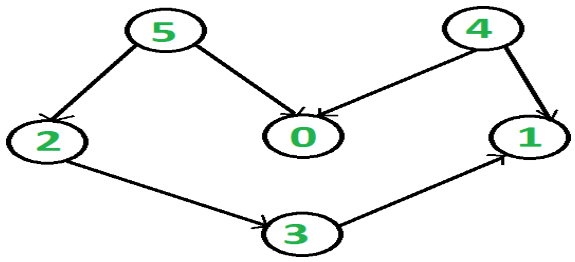
Obtain the Topological ordering of vertices in a given digraph.

2. Apparatus:

- Text Editor
- Laptop/ PC with C++ compiler.

3. Algorithm/Theory

We recommend first seeing the implementation of DFS. We can modify DFS to find Topological Sorting of a graph. In DFS, we start from a vertex, we first print it and then recursively call DFS for its adjacent vertices. In topological sorting, we use a temporary stack. We don't print the vertex immediately, we first recursively call topological sorting for all its adjacent vertices, then push it to a stack. Finally, print the contents of the stack. Note that a vertex is pushed to stack only when all of its adjacent vertices (and their adjacent vertices and so on) are already in the stack. Below image is an illustration of the above approach:

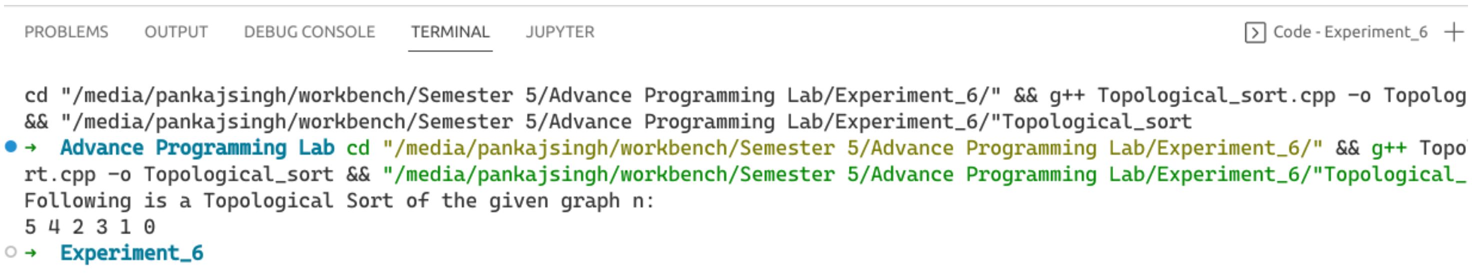


4. Program/Code

```
#include <iostream>
#include <list>
#include <stack>
using namespace std;
class Graph {
int V;
list<int>* adj;
void topologicalSortUtil(int v, bool visited[], stack<int>& Stack);
public:
Graph(int V);
void addEdge(int v, int w);
void topologicalSort();
};
Graph::Graph(int V)
{
this->V = V;
adj = new list<int>[V];
}
void Graph::addEdge(int v, int w)
{
adj[v].push_back(w);
}
void Graph::topologicalSortUtil(int v, bool visited[],
stack<int>& Stack)
{
visited[v] = true;
list<int>::iterator i;
for (i = adj[v].begin(); i != adj[v].end(); ++i)
if (!visited[*i])
topologicalSortUtil(*i, visited, Stack);
Stack.push(v);
}
void Graph::topologicalSort()
{
```

```
stack<int> Stack;
bool* visited = new bool[V];
for (int i = 0; i < V; i++)
visited[i] = false;
for (int i = 0; i < V; i++)
if (visited[i] == false)
topologicalSortUtil(i, visited, Stack);
while (Stack.empty() == false) {
cout << Stack.top() << " ";
Stack.pop();
}
}
int main()
{
Graph g(6);
g.addEdge(5, 2);
g.addEdge(5, 0);
g.addEdge(4, 0);
g.addEdge(4, 1);
g.addEdge(2, 3);
g.addEdge(3, 1);
cout << "Following is a Topological Sort of the given graph n: \n";
g.topologicalSort();
cout<<endl;
return 0;
}
```

6. Output



```
cd "/media/pankajsingh/workbench/Semester 5/Advance Programming Lab/Experiment_6/" && g++ Topological_sort.cpp -o Topolog
&& "/media/pankajsingh/workbench/Semester 5/Advance Programming Lab/Experiment_6/"Topological_sort
● → Advance Programming Lab cd "/media/pankajsingh/workbench/Semester 5/Advance Programming Lab/Experiment_6/" && g++ Topo
rt.cpp -o Topological_sort && "/media/pankajsingh/workbench/Semester 5/Advance Programming Lab/Experiment_6/"Topological_
Following is a Topological Sort of the given graph n:
5 4 2 3 1 0
○ → Experiment_6
```

7. Learning Outcomes:

1. Learned the concepts of Topological sorting

2. Learned the concepts of Graphs.

3. Learned to write a program for the above problem.

4. Learned to use vs code effectively.

Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.			
2.			
3.			