Assignment 6 : DFS and Topological Sort

**Deadline: 19/01/2021 11:55 PM**

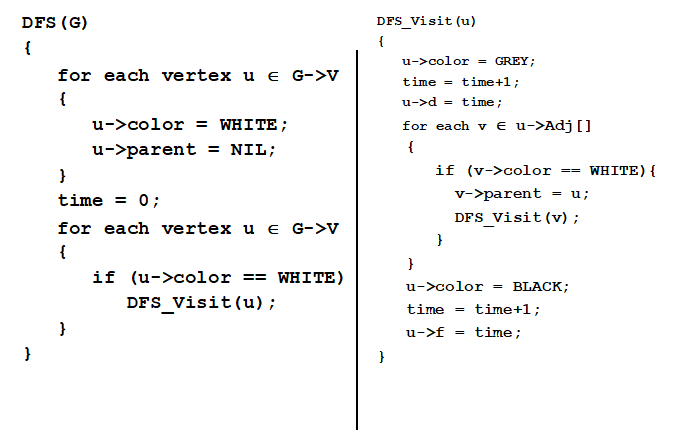
**Instruction:**

1. Write your code in the c file named “DFS.cpp” and “Topological Sort.cpp”.
2. Avoid plagiarism. If you are found to adopt any unfair means you will get a straight 0.
3. Zip the cpp files and upload the zip file in elms.
4. Deadline is 19/01/2021 11:55 PM.

**Task:**

In this assignment you will have to implement DFS and Topological Sort. Basic operations of the Graph data structure is given in the cpp files.

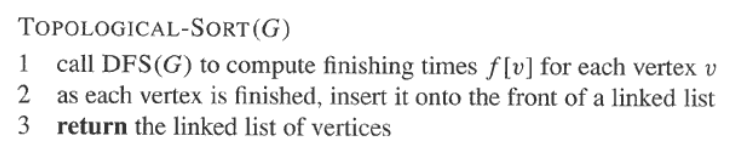
In “DFS.cpp” file you have to complete **DFS** and **DFS\_visit** functions. Psuedocode of DFS and DFS\_visit is given below.



**Important:**

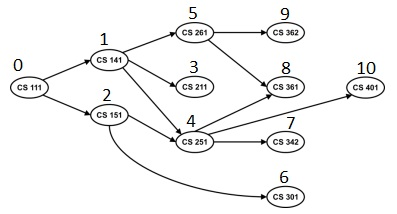
This is a pseudocode and it is obvious that it does not resemble C++ syntax. You have to complete the code in C++. You can refer to BFS fucntion if you find any difficulty understanding how to access the arrays and variables. I have made necessary arrangemens so that you can pass necessary variables and arrays as parameters from DFS fuction to DFS\_visit function. So, it should be convenient to access the variables and arrays from both functions. Whenever you make a call to DFS\_visit, make sure you pass the necessary parameters.

Now, in “Topological sort.cpp” you will need a linked list to add the finished vertices to get the topologically sorted order. In order to do that, you have to pass the linked list to **DFS\_topo** and **DFS\_visit\_topo** functions. The algorithm for Topological sort is given below.



Make sure you pass the linked list as well as other parameters when you make a call to **DFS\_visit\_topo.** Rest you should be albe to figure out by yourself.

Now we want to apply topological sort to the following graph. The vertices in the graph represents some courses (like CS111, CS141, CS151 etc). This is a Directed acyclic graph (DAG). Here an incoming edge to vertex “CS151” from vertex “CS111” means that CS111 is prerequisite course for CS151. Similarly CS141 and CS151 are prerequisite courses for CS251. There are no prerequisite courses for CS111 as vertex “CS111” has no incoming edges. We want to find an order in which we can take the courses without violating the prerequisite constraint. Topological sort is the appropriate algorithm for that.



If you go through the code in “Topological sort.cpp” you will see that the code is already set up according to this graph. Once you complete **DFS\_topo** and **DFS\_visit\_topo** functions it should be able to correctly determine the precedence order.

**Caution:**

As this assignment will be counted as the Final of the course, sharing code is strictly prohibited. If found guilty, both parties will be penalized.