

# CS-310, Fall 2021

## Assignment 2

Assigned: Oct. 15, Due: Friday, October. 22, 11:55PM

October 14, 2021

**Total Points = 60**

**Problem 1. (20 points)** Suppose that you have been hired by the CS Department to prepare the course offering schedule of CS courses for the next 5 years. You are given the following scheduling constraints:

1. All CS courses are offered in regular semester only, i.e., in Fall semester or in Spring semester. No CS course is offered in the summer semester.
2. Due to shortage of faculty, same course cannot be offered in both Fall and Spring. A course is offered either in Fall semester or in Spring semester but not in both semesters.
3. All CS courses, except *CS100*, have one or more prerequisites. *CS100* does not have any prerequisite. Any given course and its prerequisite course(s) cannot be offered in the same semester. We refer to a pair of courses  $(c_i, c_j)$  as conflicting courses if  $c_i$  is a prerequisite of  $c_j$ . Note that the conflict relation is not a transitive relation. For example *CS100* is a prerequisite of *CS200* and *CS200* is a prerequisite of *CS300*. However, *CS100* and *CS300* are not conflicting courses because *CS100* is not a direct prerequisite of *CS300*.

Suppose that there are  $n$  CS courses and  $r$  pair of conflicting courses. Give an  $O(n+r)$  time algorithm that returns a feasible course offering schedule, including all  $n$  courses subject to the above constraints. Your algorithm should return “FALSE” if no feasible schedule exists

**Problem 2. (10 points)** Explain how a vertex  $u$  of a directed graph can end up in a depth-first tree containing only  $u$ , even though  $u$  has both incoming and outgoing edges in  $G$ .

**Problem 3. (10 points)** Consider a directed graph  $G = (V, E)$ , where

$V = \{a, b, c, d, e, f, g, h\}$ , and

$E = \{(a, e), (a, h), (b, c), (b, d), (c, d), (d, a), (d, e), (e, f), (f, h), (g, f), (h, e), (h, g)\}$

Find the strong components of  $G$ .

**Problem 4. (10 points)** Consider a directed graph  $G = (V, E)$ , where

$V = \{a, b, c, d, e, f\}$ , and

$E = \{(a, b), (a, d), (b, c), (c, f), (d, e), (e, f)\}$

How many topological orderings does  $G$  have? Write all possible topological orderings.

**Problem 5. (10 points)** The topological ordering algorithm discussed in class repeatedly finds a node with no incoming edges and deletes it. This will eventually produce a topological ordering, provided that the input graph is a DAG.

Extend the topological ordering algorithm so that, given any arbitrary directed graph  $G$ , it outputs one of the two things: (a) a topological ordering, thus establishing that  $G$  is a DAG; or (b) a cycle in  $G$ , thus establishing that  $G$  is not a DAG. The running time of your algorithm should be  $O(m + n)$  for a directed graph with  $n$  nodes and  $m$  edges.