

# Homework Assignment No. 5

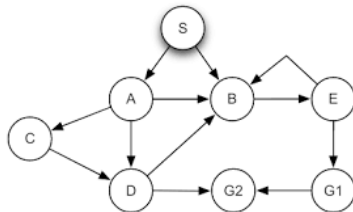
CSCI 4470/6470 Algorithms, CS@UGA, Fall 2019

**Due Monday November 4, 2019**

*Five questions with total 120 points.*

**Homework submission should be in hardcopy. If for a reason an email submission is necessary, it needs an approval from this instructor. Submission needs to be received by 5:00pm on the due day.**

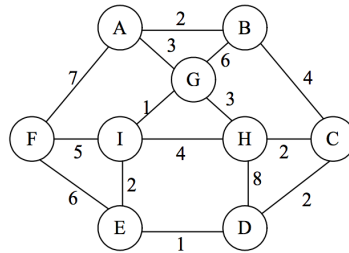
1. **(20 points)** Consider applying the DFS algorithm on the following directed graph. The algorithm colors and labels discover time and finish time for vertices in the graph. We assume the search starts from vertex  $S$  and neighbors are explored in the alphabetical order.



- (1) Show the colors of nodes in the graph when vertex  $D$  is just being colored gray;
  - (2) Show the discover time and finish time for every vertex in the graph after the DFS algorithm terminates.
2. **(30 points)** Design a DFS-based algorithm that, given the input of a directed graph  $G = (V, E)$  and a vertex  $v \in V$  in the graph, finds whether  $v$  is contained in a directed cycle in  $G$  (i.e., if there is a directed cycle in  $G$  the includes  $v$ ). For this question, (1) Outline the idea of your algorithm; (2) Write in pseudo code (not in any programming

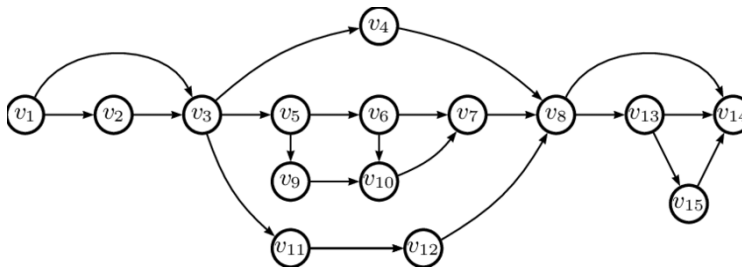
language) your full algorithm. You may assume that the input graph is stored in an adjacency list *adj*.

3. (20 points) Prove that the Prim's algorithm is based on **some greedy choice property** associated with the MST problem.
4. (20 points) Run the Prim's and Kruskal's algorithm on the following weighted graph and show the content of set  $A$  as results of the two algorithms, respectively, when  $A$  contains exactly 6 vertices. You may assume the Prim's begins from vertex  $F$ .



5. (30 points) Consider the **All-Pairs Reachability** problem: given a directed graph  $G = (V, E)$ , computes  $r(u, v)$  for every pair of vertices  $u, v \in E$  such that  $r(u, v) = 1$  if there is a directed path from  $u$  to  $v$ , and  $r(u, v) = 0$  otherwise.

On directed acyclic graphs (DAGs) (see the following example), due to the precedence of vertices, the **Reachability** problem can be solved with dynamic programming.



- (1) Formulate a recursive solution for objective function  $r(u, v)$ , using the given information of edges  $E$ ;
- (2) Give base cases for  $r(u, v)$ ;

- (3) Derive the worst case time complexity (upper bound) based on the dynamic programming idea that you have.

*The answers must be word-processed or typed. You may substitute formulae and figures with hand-writings. Your submitted algorithms should be in the pseudo-code, not in any specific programming language. Answers deviating from these requirements will be returned without grading.*

*The answers must be the student's own work. Idea sharing and referencing to others' work (including those online) are not allowed. Plagiarism and other forms of academic dishonesty will be handled within the guidelines of the Student Handbook and reported to the University.*