CSE317 Design & Analysis of Algorithms

First Term Examination – Fall'17

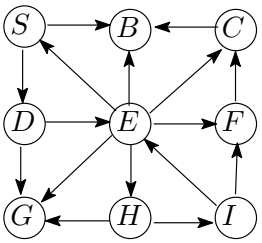
Max Marks: 25 Duration: 3 hours

Answer all questions. Use seperate answer sheet. Be to the point. Show your work.
Please give clear and rigorous answers.

Name: _____ **ERP:** _____

Question 1: Graph Search ..... *6 marks*

(a) [4 marks] Consider the following directed graph pictured below right.



i. Perform DFS search starting from vertex S. Indicate pre/post number of each vertex and draw non-tree edges as dotted lines.

ii. Perform BFS search starting from vertex S. Indicate dist value of each vertex and draw non-tree edges as dotted lines.

(b) [2 marks] Modify DFS algorithm to check whether a given graph contains a cycle of odd length or not.

Question 2: Asymptotic Notation ..... *8 marks*

(a) [4 marks] For each pair of expressions $T(n)$ and $f(n)$ below, indicate whether $T(n)$ is O, Ω, or Θ of $f(n)$.

$T(n)$	$f(n)$	$T(n) \stackrel{?}{=} O(f(n))$	$T(n) \stackrel{?}{=} \Omega(f(n))$	$T(n) \stackrel{?}{=} \Theta(f(n))$
10	$\log 10$			
$n \log n$	$n\sqrt{n}/2$			
$n2^n$	3^n			
$n!$	n^n			

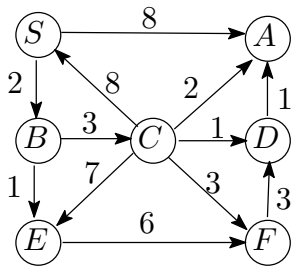
(b) [2 marks] Prove that $n^2 = O(2^n)$.

(c) [2 marks] Why can we write $O(m \log n)$ instead of $O(m \log m)$, if n and m denotes the number of nodes and edges, respectively, in a graph?

Question 3: Shortest Paths ..... *5 marks*

(a) [3 marks]

Find shortest path tree rooted at S in the weighted graph shown on the right using Dijkstra algorithm. Give a table showing the vertex removed in each iteration and `dist[]` values after the iteration is completed. Also draw the shortest-path tree.

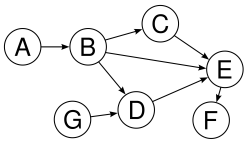


(b) [2 marks] Give an example of weighted graph where Dijkstra algorithm fails, i.e., the resulting `dist` values are incorrect.

Question 4: Applications of DFS ..... *6 marks*

(a) [2 marks]

Find a *linear ordering* (also called *topological sorting*) of directed acyclic graph shown on right using the algorithm discussed in class.



(b) [2 marks] Given a directed acyclic graph G, describe a linear-time algorithm to determine whether there exists a vertex that can be reached by every other vertex.

(c) [2 marks] True of False? Briefly justify your answer.

i. A shortest path in a graph remains shortest after incrementing weight of each edge by 1.

ii. Given a connected graph $G = (V, E)$, if a vertex v is visited during level k of a breadth-first search from source vertex s, then every path from s to v has length at most k.