Name: Daniel Kim
ID: 102353420

CSCI 3104, Algorithms — **Profs. Chen & Grochow**
Final Exam S12–S15 — **Spring 2020, CU-Boulder**

Instructions: This quiz is open book and open note. You **may** post clarification questions to Piazza, with the understanding that you may not receive an answer in time and posting does count towards your time limit. Questions posted to Piazza **must be posted as PRIVATE QUESTIONS.** Other use of the internet, including searching for answers or posting to sites like Chegg, is strictly prohibited. Violations of these are grounds to receive a 0 on this quiz. Proofs should be written in **complete sentences. Show and justify all work to receive full credit.**

TIMING: If you are not attempting all the standards in a given quiz, please only use the ordinary amount of time for the number of standards you attempt. For example, if you are only attempting one standard on a 4-standard quiz, please only use 30 min (or 38 for 1.5x, 45 for 2x).

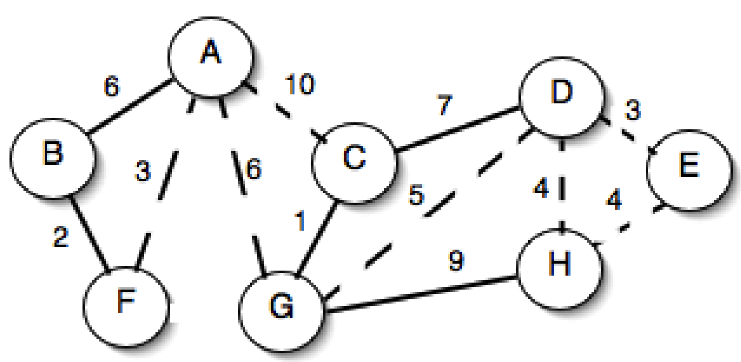
YOU MUST SIGN THE HONOR PLEDGE. Your quiz will otherwise not be graded.
Honor Pledge: On my honor, I have not used any outside resources (other than my notes and book), nor have I given any help to anyone completing this assignment.

Your Name: Daniel Kim ______________________________

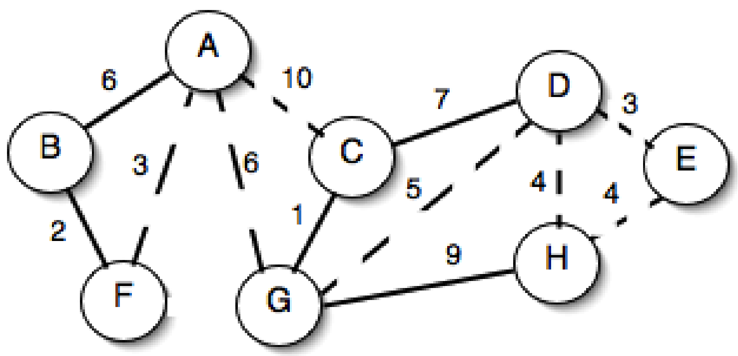
Quicklinks: 12 13 14 15 15a 15b

---

12. **Standard 12.** Consider the following graph G, consisting of both dashed and undashed edges. The solid edges form a forest, as would occur in the middle of a algorithm to find a minimum spanning tree. In the context of trying to find an MST, for all of the dashed edges, list which ones are safe, which are useless, and which are neither safe nor useless.



(BEGIN YOUR ANSWER ON THE NEXT PAGE.)

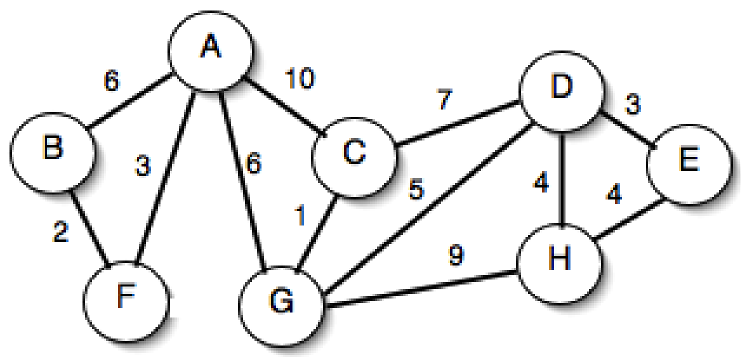


To begin, we first study the edge (A, F). We can conclude that the edge (A, F) is useless because both of the edge's endpoints are in the same part of the forest. In addition, edge (D, G) and edge (D, H) are also useless due to the same reason. Moving on, we consider study edge (A, G), and (D, E). In the forest ABF, the edge (A, G) is a safe edge because the edge (A, G) is the minimum weight edge with exactly one endpoint. In the forest CDGH, the edge (D, E) is also a safe edge because the edge (D, E) is the minimum weight edge with exactly one endpoint. Lastly, edge (A, C) and edge (H, E) is neither safe nor useless.

Name: Daniel Kim
ID: 102353420

CSCI 3104, Algorithms **Profs. Chen & Grochow**
Final Exam S12–S15 **Spring 2020, CU-Boulder**

13. **Standard 13.** Consider the following graph G.



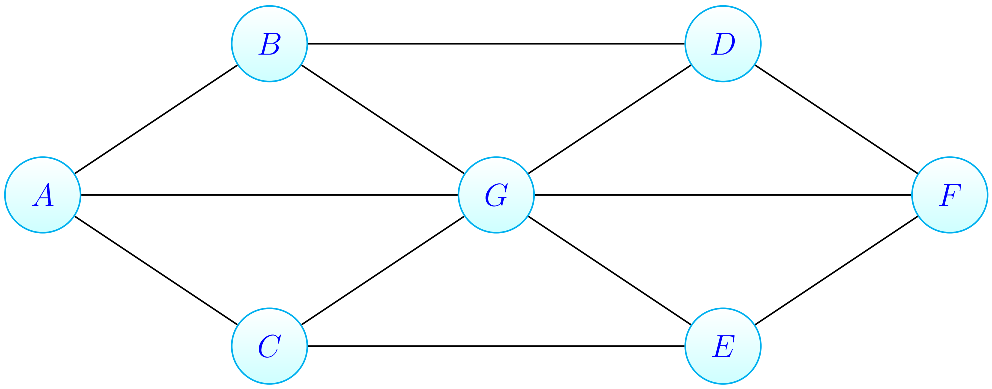
We seek to construct a minimum spanning tree from G, using **Prim's Algorithm**, starting with the vertex G. Determine the first six edges selected to be included in the MST. Clearly articulate the steps Prim's Algorithm takes to select these edges.

(BEGIN YOUR ANSWER ON THE NEXT PAGE.)

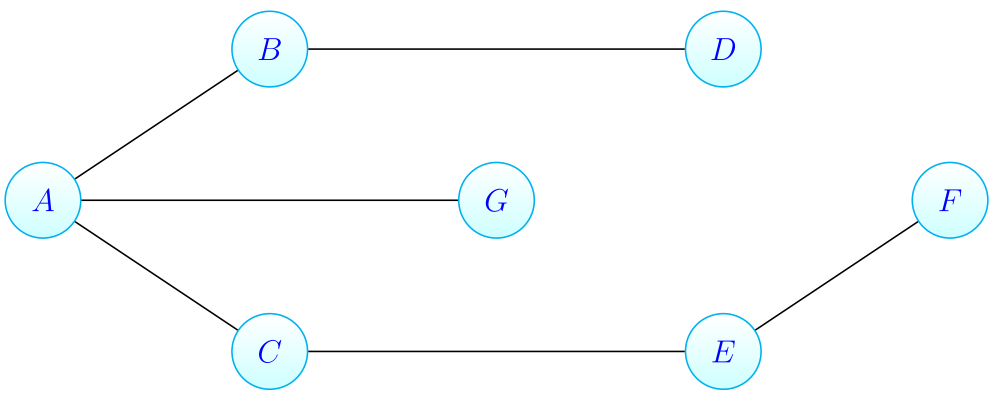
Since vertex G is selected as the starting point, the graph G is composed of visited set $\{G\}$ and rest of the unvisited vertices. Per Prim's algorithm, at each step, we add the edge with the smallest weight that connects the visited set $\{G\}$ to the rest of the unvisited vertices. So the first edge we will add to $\{G\}$ is (C, G) because it has the smallest weight among all the vertices that connect $\{G\}$. Then, we mark vertex C as visited, so the visited set is now $\{G, C\}$. Next, we find the edge that has the smallest weight and it also connected to $\{G, C\}$. And we find the edge (D, G) with weight 5, which is the smallest. So we mark vertex D as visited, so the visited vertices are now $\{G, C, D\}$. Next, we find the edge that has the smallest weight and it also connected to $\{G, C, D\}$. And we find the edge (D, E) with weight 3, which is the smallest. So we mark vertex E as visited, so the visited vertices are now $\{G, C, D, E\}$.Next, we find the edge that has the smallest weight and it also connected to $\{G, C, D, E\}$. And we find the edge (E, H) with weight 4, which is the smallest. So we mark vertex H as visited, so the visited vertices are now $\{G, C, D, E, H\}$. Next, we find the edge that has the smallest weight and it also connected to $\{G, C, D, E, H\}$. And we find the edge (A, G) with weight 6, which is the smallest. So we mark vertex A as visited, so the visited vertices are now $\{G, C, D, E, H, A\}$. Finally, we find the edge that has the smallest weight and it also connected to $\{G, C, D, E, H, A\}$. And we find the edge (A, F) with weight 3, which is the smallest. So we mark vertex F as visited, so the visited vertices are now $\{G, C, D, E, H, A, F\}$. Therefore, the next six edges are (G, C), (G, D), (D, E), (E, H), (G, A), (A, F).

Name: Daniel Kim
ID: 102353420

14. **Standard 14.** Consider the following graph with source node A:



Is it possible to obtain the following tree using BFS? If so, explain the steps taken by BFS to yield this tree; if not, clearly justify why not.

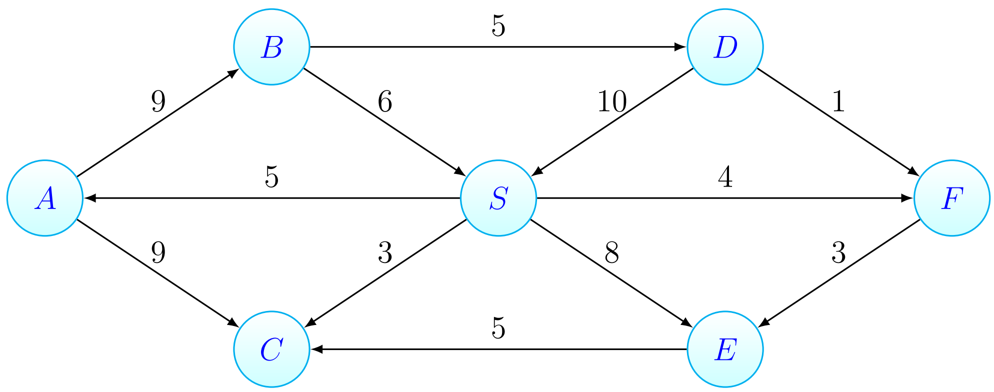


(BEGIN YOUR ANSWER ON THE NEXT PAGE.)

YOUR ANSWER HERE FOR STANDARD 14. (YOU CAN DELETE ALL THIS TEXT IN CAPS.)

IF YOU ARE HANDWRITING AND INSERTING AN IMAGE, SEE THE COMMENTED CODE BELOW IN THE .TEX FILE. PLEASE BE SURE TO ROTATE YOUR IMAGE TO THE CORRECT ORIENTATION (CAN BE DONE IN THE LATEX DIRECTLY; SEE COMMENTS.)

15. **Standard 15.** Consider the following directed, weighted graph G. At the first iteration of Dijkstra's Algorithm, using S as the source vertex, we examine the edges $(S, A), (S, C)$, (S, E), and (S, F) edges by placing them into a priority queue. However, only (S, C) is **selected** at the first iteration.



(Questions begin on next page. THIS QUESTION HAS TWO PARTS, (a) AND (b).

(a) What are the next five edges **selected** by Dijkstra's algorithm? Show your work. (BEGIN YOUR ANSWER ON THE NEXT PAGE.)

$(S, F), (S, A), (F, E), (A, B), (B, D)$ are the next five edges.

To begin, (S, C) is selected at the first iteration from the given problem. We update the distance for the source to 3 for node C. In addition, since there are no unvisited neighbors of C, we can mark node C as visited. For next node, we need to find the unvisited node with the smallest minimum distance which is node F. Now, ignoring the visited nodes, we can check the unvisited neighbors of node F which concludes with checking node E. We update the distance to the source from infinity to 7 for node E. After that, we mark node F as visited, and pick the next current node which is node A. Again, ignoring the visited nodes, we now check for the unvisited neighbors of node A which concludes with checking node B. We update the distance to the source from infinity to 14 for node B. Then, we mark node A as visited and pick the next current node which is node E. Again, ignoring the visited nodes, we now check for the unvisited neighbors of E, but they are none. Because of that we mark node E as visited and pick the next current node which is node B. Again, ignoring the visited nodes, we now check for the unvisited neighbors of node B which is node D. We update the distance to the source from infinity to 19 for node D.

(b) After these have been selected, what are the distances from S that the algorithm has recorded for each vertex in G?
(BEGIN YOUR ANSWER ON THE NEXT PAGE.)

d(S, F) = 4
d(S, A) = 5
d(S, E) = 7
d(S, B) = 14
d(S, D) = 19