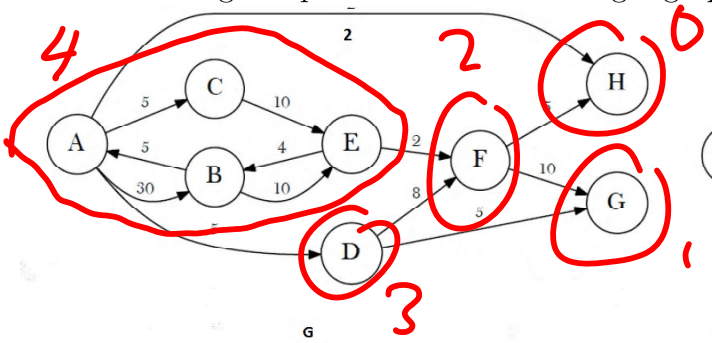
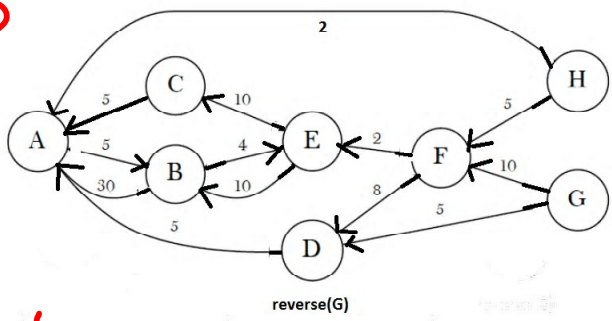
COS 226 Data Structures and Algorithms
Spring 2016 - Flipped Lecture Handout - 3/28/16

DiGraph and MST

1. **Strong Components**
   Find the strong components of the following digraph.



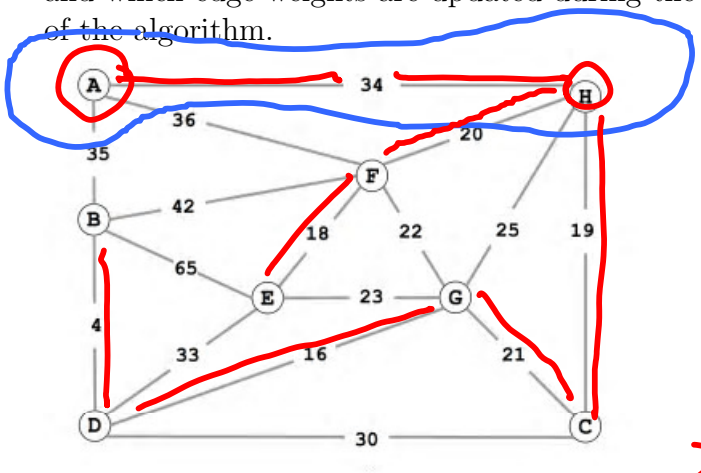
G



reverse(G)

HGFDABEC

2. **Prims Eager version**
   Find the MST of the following graph using Prim's eager algorithm. Create a table with 3 columns named, vertex, edge, weight to show which vertices are in the MST and which edge weights are updated during the execution of the Prim's eager version of the algorithm.



vertex	edge	weight
A	A-H	34
B	A-B	35
C	H-C	19
D		
E		
F	~~A-F~~ H-F	~~36~~ 29
G	H-G	25
H		

complete the rest of the table

decrease key operation

3. **Graph Interview Questions**

Suppose you are applying for a job at a software company. You are asked to classify if the the following tasks are possible, impossible or unknown. You will need to justify your answers to the interviewer

(a) Given an undirected graph, determine if there is a path of length V-1 with no repeated vertices in worst case time EV

This is s hard problem. No polynomial time solution known

(b) Given a digraph, determine if there is a path of between every pair of vertices in time proportional to E+V

Apply the Kosaraju Algorithm to see if there is one SCC. Then the answer is yes.

(c) Given an undirected graph, determine if there is a path of between every pair of vertices in time proportional to E+V

Look for one connected component. The answer is yes.

(d) Given a digraph, determine if the digraph is a rooted DAG in time proportional to E+V. Rooted DAG means it is a directed acyclic graph, and there is a unique vertex with outdegree equal to 0.

Determine if the graph is acyclic - E+V. Then find a vertex with zero outdegree

(e) Given an airline route map, find the minimum number of connections from a given city to every other city in linear time.

determine the shortest path tree using BFS staring with any vertex

(f) If you have a choice between Kruskals and Prim's algorithm for finding MST's which one would you pick and why?

Kruskal's uses extra memory (PQ of edges) than Prim's. I prefer prims

4. **Graph Questions - Challenging**

(a) Suppose you know the MST of a weighted graph G. A new graph G_0 is formed by adding a new edge v-w of weight c is to graph G. You may assume all edge weights are distinct. Design an algorithm to determine if the MST in G is also an MST in G_0. What is the order of growth of your algorithm?

MST of the graph connects all vertices. Now find a path from v to w in the MS in time V. If all edges in the path > c, then MST of G is the same as MST of G If not, you can remove an edge > c from the path and replace the edge with tl new edge. In this case they are not the same.

(b) Design an algorithm to find a vertex whose removal will not disconnect the graph. What is the order of growth of your algorithm?

Lets assume this is a connected graph. If we do BFS and build the shortest path tree E + V) time, we can find a leaf node in the tree that can be removed w/o disconnectin the entire graph

(c) Design an algorithm to find the shortest cycle in a directed graph? What is the order of growth of your algorithm?

Start with any vertex v. Build the shortest path graph from v to every other node. For all other vertice x, if the edge (x,v) exists, then calculate the cycle length using shortest path length + 1. find the minimum such thing. We will have to do this for all vertices. So the order of growth is EV