http://algs4.cs.princeton.edu

Flipped lecture 5

- *Directed graphs*
- *MSTs*

Directed graphs

Questions

- We used DFS to find all the states reachable from a source vertex.
  - Would BFS work? Why or why not?
- Identify a situation where you need to use BFS instead of DFS.
- Identify a situation where you need to use DFS instead of BFS.

(b) Consider two vertices x and y that are simultaneously on the FIFO queue at some point during the execution of breadth-first search from s in an undirected graph. Which of the following are true?

I. The number of edges on the shortest path between s and x is at most one more than the number of edges on the shortest path between s and y.

II. The number of edges on the shortest path between s and x is at least one less than the number of edges on the shortest path between s and y.

III. There is a path between x and y.

(a) I only.

(b) I and II only.

(c) I and III only.

(d) I, II and III.

(e) None.

(b) Consider two vertices x and y that are simultaneously on the function-call stack at some point during the execution of depth-first search from vertex s in a *digraph*. Which of the following must be true?

I. There is *both* a directed path from s to x *and* a directed path from s to y.

II. If there is *no* directed path from x to y, then there is a directed path from y to x.

III. There is *both* a directed path from x to y *and* a directed path from y to x.

(a) I only.

(b) I and II only.

(c) I and III only.

(d) I, II and III.

(e) None.

Graph problem (Final, Spring 2013)

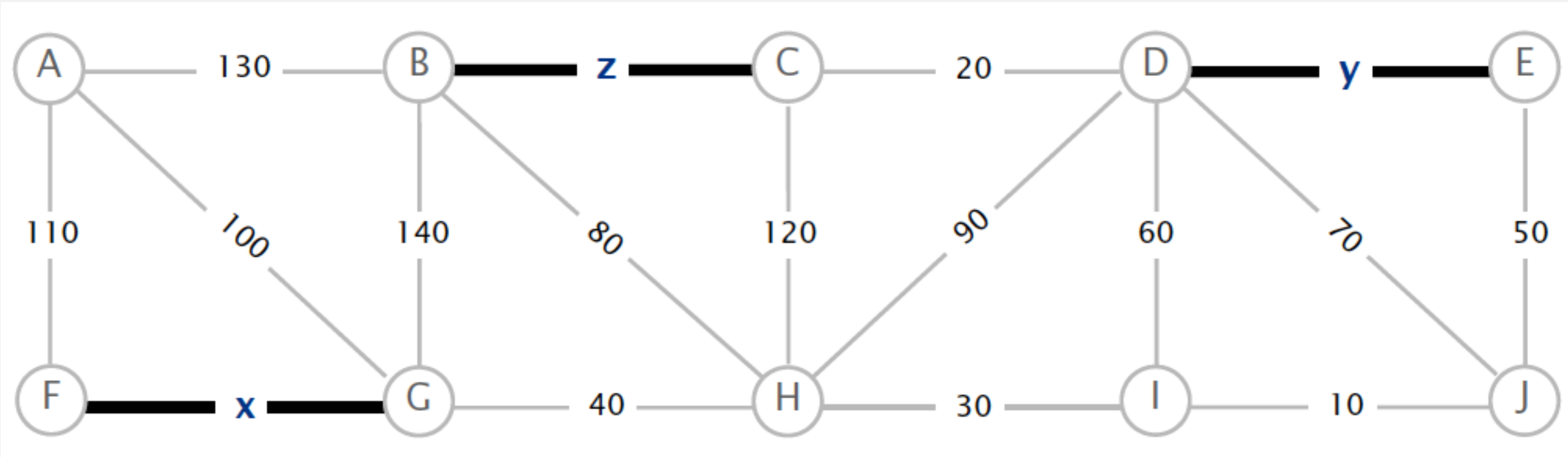
- Let G=(V,E) be an unweighted, directed graph.
- Let s and t be two vertices of G.
- Suppose we want an algorithm that finds all **distinct shortest** paths from s to t.
  - Distinct paths may share some but not all edges.
- You may assume there are no parallel or self loops.

Critique the following solution (example on board)

- Run BFS, and mark each node with a distance and a counter.
- When a node is dequeued, for each neighbor:
  - If that neighbor is unmarked, set distance to self.distance + 1 and set counter to 1 and enqueue.
  - If that neighbor is marked, and distance is equal to self.distance + 1, increment counter by 1, but don't enqueue.
  - If that neighbor is marked and distance is > self.distance + 1, ignore

B level MST

Suppose the that the MST of the graph below contains the edges with weights x, y, and z.



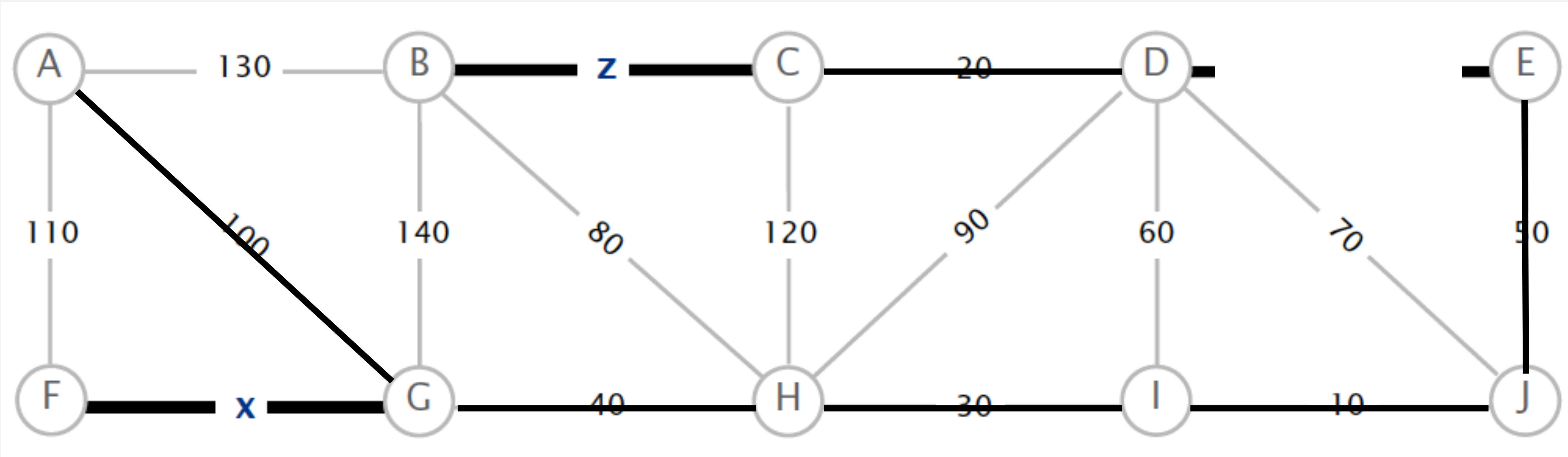
- True or false: The minimum weight edge from every node must be part of the MST.
- List the weights of the **other** edges in the MST:

  _10_ ____ ____ ____ ____ ____

- What are the possible values for the weights of x, y, and z?

B level MST

Suppose the that the MST of the graph below contains the edges with weights x, y, and z.

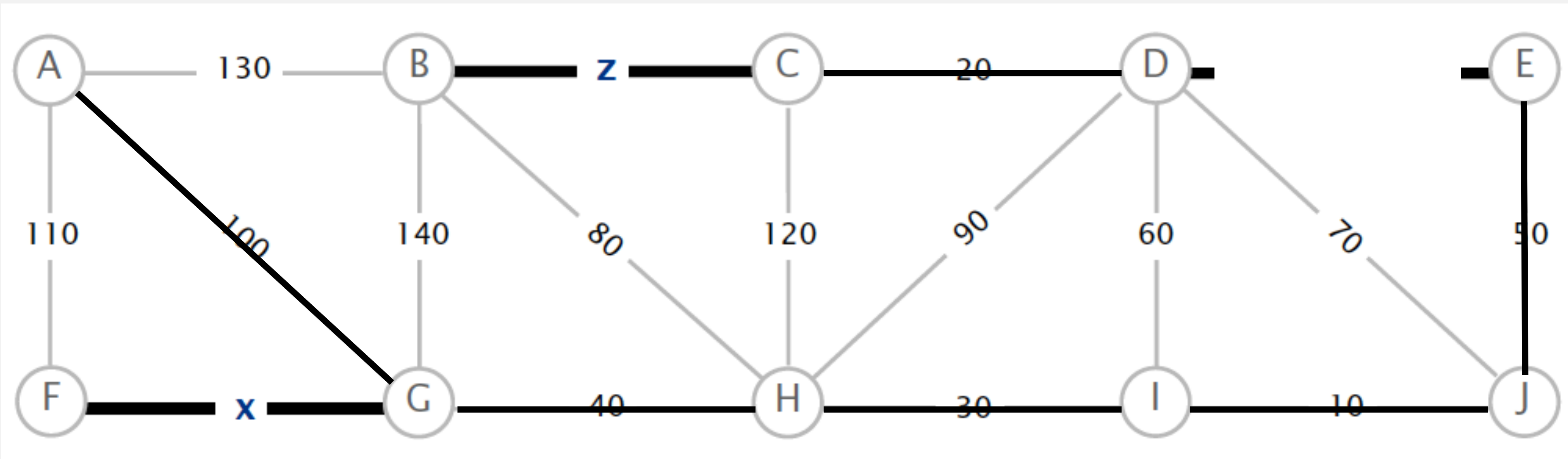


- True or false: The minimum weight edge from every node must be part of the MST - true by cut property!
- List the weights of the **other** edges in the MST:

  _10 _30 _50 _20 _40 100
- What are the possible values for the weights of x, y, and z?
  - x <= 110, y <= ?

B level MST

Suppose the that the MST of the graph below contains the edges with weights x, y, and z.

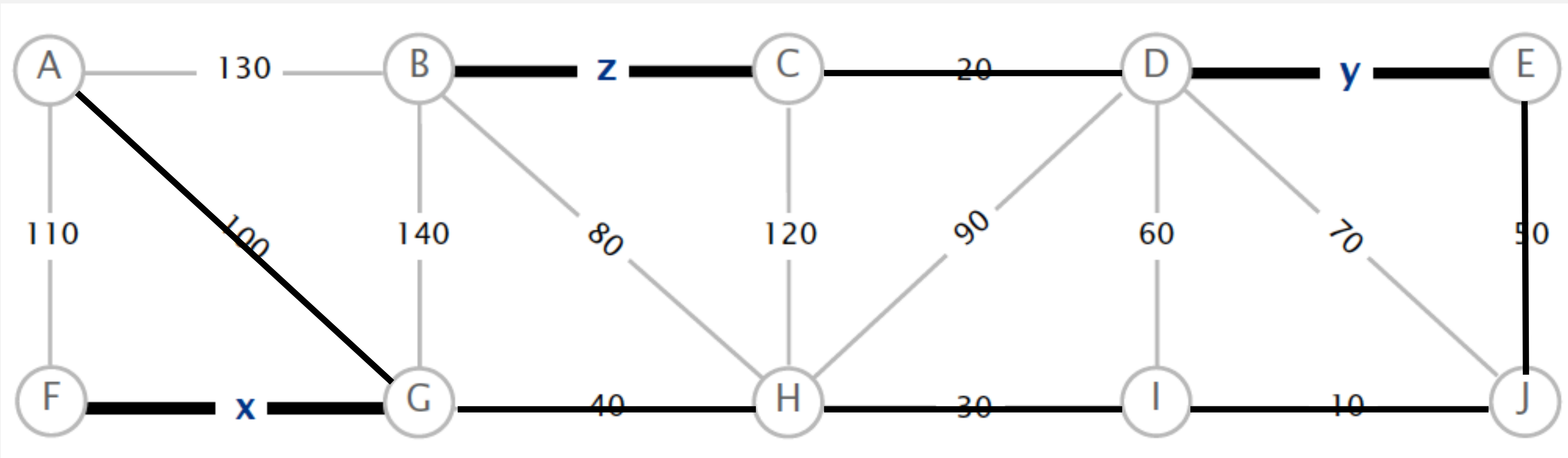


- True or false: The minimum weight edge from every node must be part of the MST - true by cut property!
- List the weights of the **other** edges in the MST:

  _10 _30 _50 _20 _40 100
- What are the possible values for the weights of x, y, and z?
  - x <= 110, y <= 60,

B level MST

Suppose the that the MST of the graph below contains the edges with weights x, y, and z.



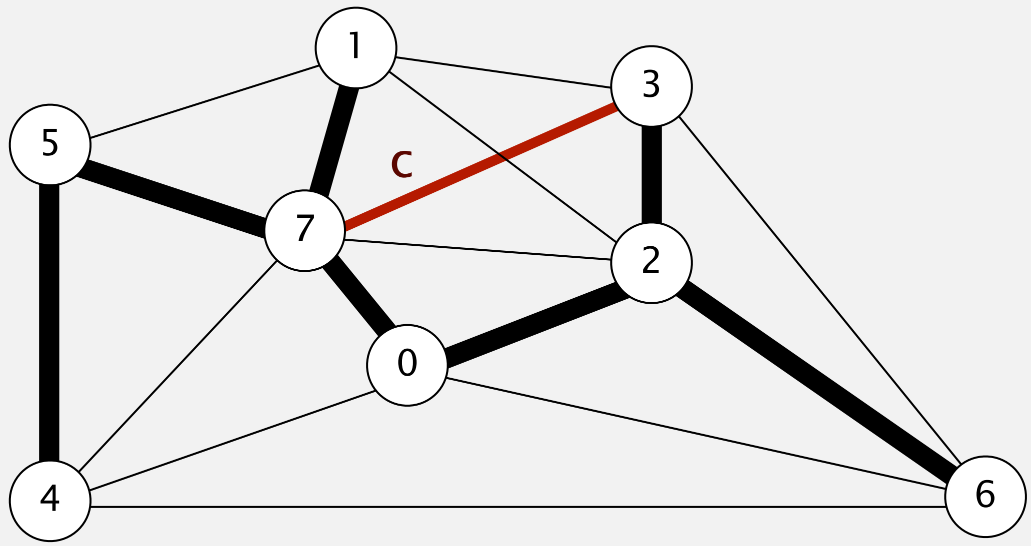
- True or false: The minimum weight edge from every node must be part of the MST - true by cut property!
- List the weights of the **other** edges in the MST:

  _10   _30   _50   _20   _40   100

- What are the possible values for the weights of x, y, and z?
  - x <= 110, y <= 60, z <= 80

A level MST

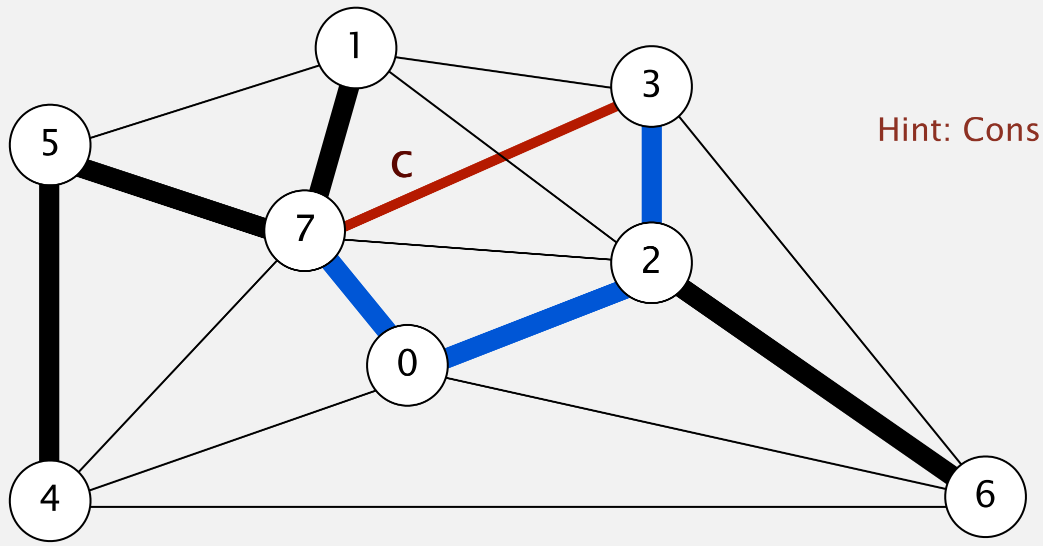
- Suppose you know the MST of G. Now a new edge v-w of weight c is added to G, resulting in a new graph G'. Design a O(V) algorithm to determine if the MST for G is also an MST for G'.



- Bonus: Given a graph G and its MST, if we remove an edge from G that is part of the MST, how do we find the new MST in O(E) time?

A level MST

- Suppose you know the MST of G. Now a new edge v-w of weight c is added to G, resulting in a new graph G'. Design a O(V) algorithm to determine if the MST for G is also an MST for G'.

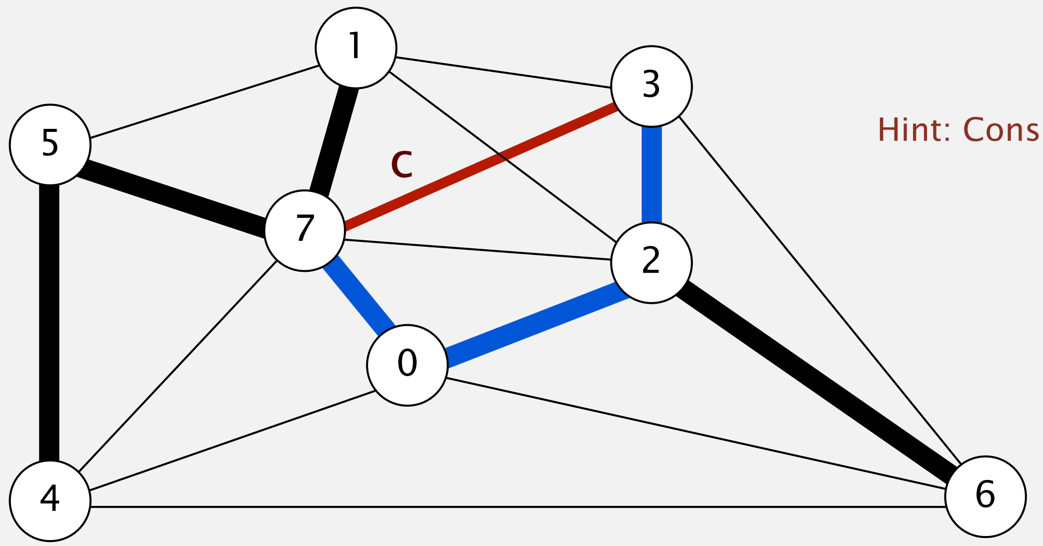


Hint: Consider the blue path.

-

A level MST

- Suppose you know the MST of G. Now a new edge v-w of weight c is added to G, resulting in a new graph G'. Design a O(V) algorithm to determine if the MST for G is also an MST for G'.



Hint: Consider the blue path.

- If any edge on the blue path is longer than c:
  - Replace that edge with c - you get a new MST with shorter distance.
- If every edge on the blue path is shorter than c:
  - Then we know original MST was the best.
- Finding the blue path: Run DFS from one of c's vertices to the other, only taking steps along the MST.