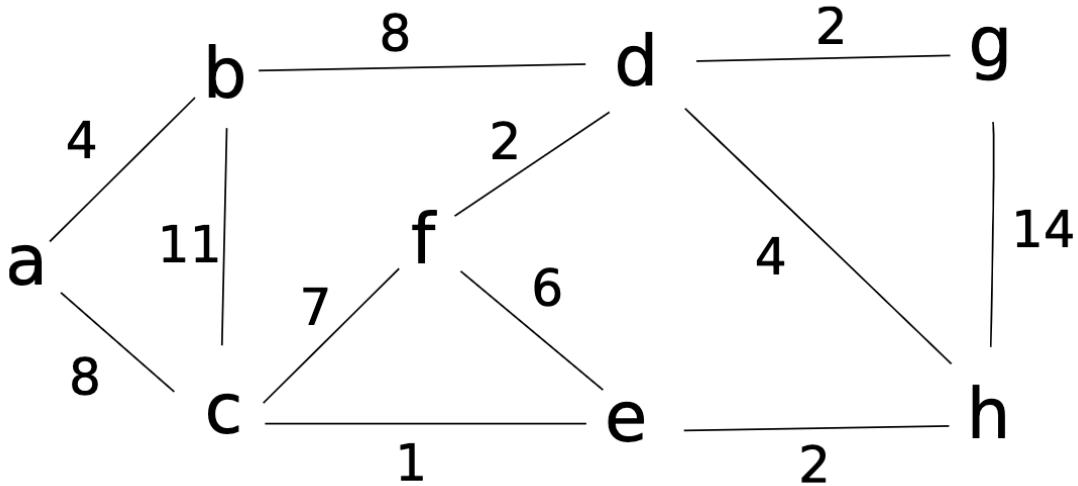


March 23, 2022

1. (pts.) Prim's

Suppose Prim's algorithm is run on the following graph, starting at node f .



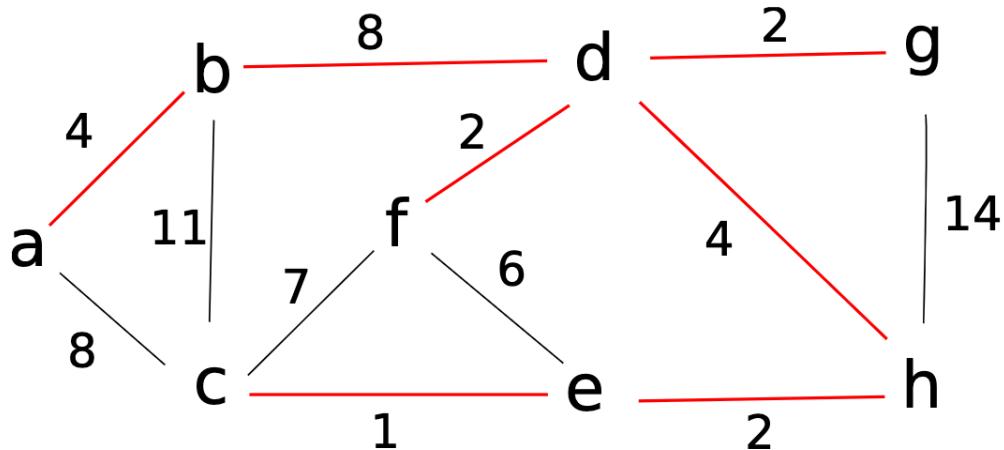
- Draw a table showing the intermediate *cost* and *prev* values of all the nodes at each iteration of the algorithm.
- Show the final *MST*.
- Run Kruskal's algorithm to find the MST of this graph. Give the edges in the order in which they are added to the MST. Use alphabetical ordering to avoid ambiguities.
- How would you find the *maximum spanning tree*?

Answer:

a)

Set S	Node							
	a	b	c	d	e	f	g	h
$\{\}$	∞/nil	∞/nil	∞/nil	∞/nil	∞/nil	$0/\text{nil}$	∞/nil	∞/nil
$\{f\}$	∞/nil	∞/nil	$7/f$	$2/f$	$6/f$		∞/nil	∞/nil
$\{f, d\}$	∞/nil	$8/d$	$7/f$		$6/f$		$2/d$	$4/d$
$\{f, d, g\}$	∞/nil	$8/d$	$7/f$		$6/f$			$4/d$
$\{f, d, g, h\}$	∞/nil	$8/d$	$7/f$		$2/h$			
$\{f, d, g, h, e\}$	∞/nil	$8/d$	$1/e$					
$\{f, d, g, h, e, c\}$	$8/c$	$8/d$						
$\{f, d, g, h, e, c, b\}$	$4/b$							
$\{f, d, g, h, e, c, b, a\}$								

b)



- c) Order in which edges are added to the Kruskal's MST: (c,e), (d,f), (d,g), (e,h), (a,b), (d,h), (b,d)
- d) Multiply the weights of all the edges by -1, and run Prim's algorithm to find the minimum spanning tree of the new graph.

2. (pts.) MST: True or False

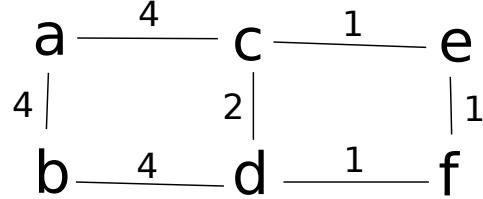
Consider an undirected and connected graph $G = (V, E)$. Do not assume that edge weights are distinct unless this is specifically stated. For the following statements, give a short explanation for why it is true or give a counterexample if it is false.

- (a) If G has more than $|V| - 1$ edges, and there is a unique heaviest edge e , then e cannot be part of an MST.

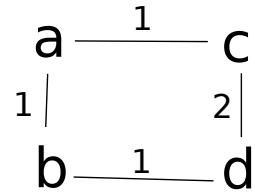
- (b) If G has a cycle with a unique heaviest edge e , then e cannot be part of any MST.
- (c) Let e be any edge of minimum weight in G . Then e must be part of some MST.
- (d) If the lightest edge in a graph is unique, then it must be part of every MST.
- (e) If e is part of some MST of G , then it must be a lightest edge across some cut of G .
- (f) If G has a cycle with a unique lightest edge e , then e must be part of every MST.
- (g) The shortest-path tree computed by Dijkstra's algorithm is necessarily an MST.
- (h) The shortest path between two nodes is necessarily part of some MST.
- (i) Prim's algorithm works correctly when there are negative edges.

Answer:

- (a) False. Consider the case where the heaviest edge is a bridge connecting two connected components of G
- (b) True. Remove e from the MST and add another edge belonging to the same cycle. Then we get a new tree with less weight.
- (c) True. e belongs to the MST produced by Kruskal.
- (d) True. If not, including e in an MST and there will be a cycle connecting the two endpoints of e . Adding e and removing another edge of the cycle produces a lighter tree.
- (e) True. Consider the cut that has u in one side and v in the other, where $e = (u, v)$.
- (f) False. Consider the following example where the middle edge (of weight 2) is the unique lightest edge of the left cycle, but it's not in the MST produced by Kruskal.



- (g) False. Consider the following example. Dijkstra's algorithm (starting from c) produces $(a, c), (a, b), (c, d)$ while Kruskal's algorithm produces $(a, c), (a, b), (b, d)$.



- (h) False. In the previous example, the shortest path between c and d is not in any MST.
- (i) True. Since negative edges don't affect the greedy choice property and the optimal substructure property. (Note, this also applies to Kruskal's algorithm.)