

1) a) A connected graph has a path between every pair of vertices.

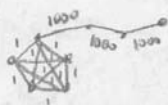


$n-1$  edges are needed to make the tree.

Using Algorithm #1, the  $m$  edges with weight 1 are picked first, then  $n-1-m$  edges with weight 1000 are picked (assuming no loops, since we're looking for min. weight a min. spanning tree can have)

$$\begin{aligned} \text{min weight} &= m \cdot 1 + (n-1-m) \cdot 1000 \quad (\text{assuming } m \leq n-1) \\ &= (n-1) \cdot 1000 \quad (\text{" " } > n) \end{aligned}$$

b)



"Use up" all weight 1 edges in a subgraph with all nodes linked to one another (a "clique")

2 nodes	1 edge
3	3
4	6
5	10
⋮	⋮

Find  $k$  such that  $\frac{k(k-1)}{2} = m$  (if no such  $k$ , then  $\frac{k(k-1)}{2} < m < \frac{k'(k'+1)}{2}$ ,  $k=k'+1$ )  
Then max weight  $= (k-1) \cdot 1 + (n-1-(k-1)) \cdot 1000$  since  $k-1$  of the length 1 edges must be used up.

2) a) Using Algorithm #1, all edges of length  $v$  are added first that do not form a cycle.

Let  $W$  be the set of all edges of length  $v$  not added since they would form a cycle.

Any member of  $W$  can be removed only if another edge not in  $W$  is added, to prevent cycles



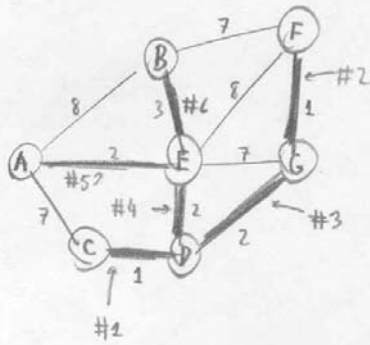
So  $v$  will not change in size, leaving the same number of edges of weight  $v$  in the graph.

b)

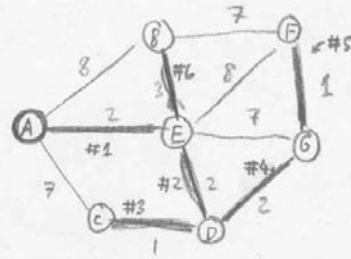


These are both SPTs. The 1st has 3 length- $v$  edges, the 2nd has 2.

3)



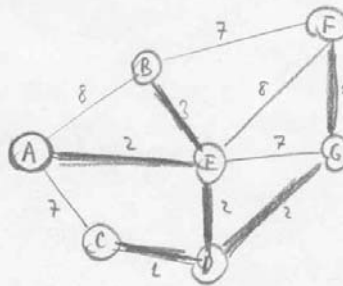
Algorithm #1



Algorithm #2

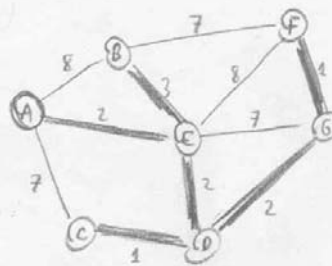
4) a) repeatedly add node with shortest path to root

- A
- AE (2)
- AED (4)
- AEDC (5)
- AEB (5)
- AEDG (6)
- AEDGF (7)



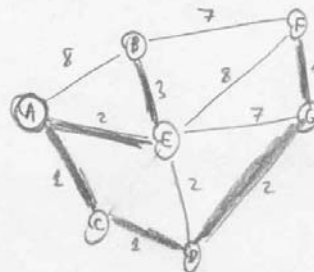
b)

	B	C	D	E	F	G
	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
EA	7A			2A		
5E			4E		10E	9E
		5D				6D
					7G	



c)

	B	C	D	E	F	G
5E	5D	4E	2A	7G	6D	
	1A					
		2C				
					4D	
						5G



d)

B	C	D	E	F	G
5E	1A	2C	2A	5G	4D
				6F	
			7G		
			8F		
		9G			
		9E			
		10G			

