Student Information

Full Name : Mohammad Mahdi Khosravi
Id Number : 2489235

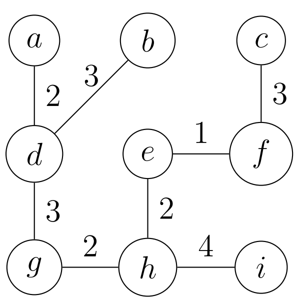
Answer 1

a)

Using Prim's algorithm.

Choice	Edge	Weight
1	{e,f}	1
2	{e,h}	2
3	{h,g}	2
4	{g,d}	3
5	{d,a}	2
6	{d,b}	3
7	{f,c}	3
8	{h,i}	4

b)



c)

This graph has a unique minimum spanning tree. However in general if a graph has an edge which its weight isn't unique, then the minimum spanning tree may not be unique because in both Prim and Kruskal's algorithm there is no specification on what to do when there are more than 1 minimum edges in a step, therefore we can make a different choice each time, leading to different minimum spanning trees.

d)

Let's assume that the unique minimum-weight edge e in a graph G is not included in an arbitrary minimum spanning tree. If we use either Prim's or Kruskal's algorithm to find a minimum spanning tree T then we do not have e as an edge in the resulting T. Also in T all the edges included are bigger than e since e is the unique minimum-weighted edge. So after the algorithm is finished, we can remove an edge from T and put e in T. This results in a new spanning tree which has a lower cost compared to the original T. This is a contradiction since T was supposed to be the minimum spanning tree whereas we showed it is not. Therefore the unique minimum edge must be in all minimum spanning trees of G.

Answer 2

G and H are isomorphic because we can define a one-to-one and onto function from G to H like this:
$f(a) = n$, $f(b) = q$, $f(c) = o$, $f(d) = r$, $f(e) = m$, $f(f) = p$
and this function preserves the adjacency of every pair of edges in G=(V,E) such that for all adjacent vertices $a, b \in V$ are adjacent to $f(a)$ and $f(b)$ are adjacent in H.

Answer 3

a

T has 7 vertices, 6 nodes and the height is 3.

b

Preorder:

Order	Vertex
1	<p:17>
2	<q:13>
3	<r:24>
4	<s:19>
5	<t:43>
6	<u:23>
7	<v:58>

Inorder:

Order	Vertex
1	<q:13>
2	<p:17>
3	<s:19>
4	<r:24>
5	<u:23>
6	<t:43>
7	<v:58>

Postorder:

Order	Vertex
1	<q:13>
2	<s:19>
3	<u:23>
4	<v:58>
5	<t:43>
6	<r:24>
7	<p:17>

c

T is a full binary tree because every internal vertex (vertex with children) has exactly 2 children.

d

T is not a complete binary tree because even though it is a full binary tree however, the leaves are not all at the same level.

e

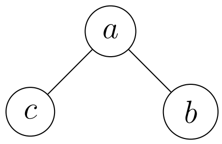
T is not a balanced binary tree because not all leaves are in level h or $h-1$, namely $< q : 13 >$ is in height 1 but the height of tree is 3.

f

For a tree to be a binary search tree, the key of each vertex must be smaller than all of the keys in the right sub-tree of the vertex and smaller than all of the keys in the right sub-tree of the vertex. T is not a binary search tree because $< r : 24 >$ is larger than $< u : 23 >$ which is a key in its right sub-tree.

g

The minimum number of nodes is $2 \times 5 + 1 = 11$.
We can prove inductively that the minimum number of nodes n for a full binary tree of height h is $n = 2h + 1$: Base case: For h=1, the minimum number of nodes is $2 * 1 + 1 = 3$.



Inductive step: Assume for height h=k, we have $n = 2k + 1$ nodes. To add an additional level we must add 2 nodes to a node because it should be a full binary tree. So from the induction hypothesis we have $n + 2 = 2k + 3 \Rightarrow n + 2 = 2(k + 1) + 1$. So the formula is proven.