PRINTABLE VERSION

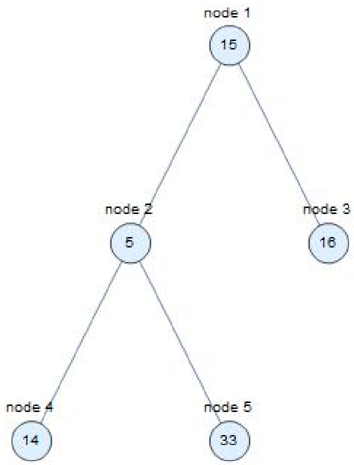
Quiz 7

You scored 100 out of 100

Question 1

Your answer is CORRECT.

A graph with nodes and edges is shown below (and we have stored random integers in each node):



Which node is displayed as the root, and what value is stored in it?

a) ○ Node 3 is the root, and its value is 16.

b) ● Node 1 is the root, and its value is 15.

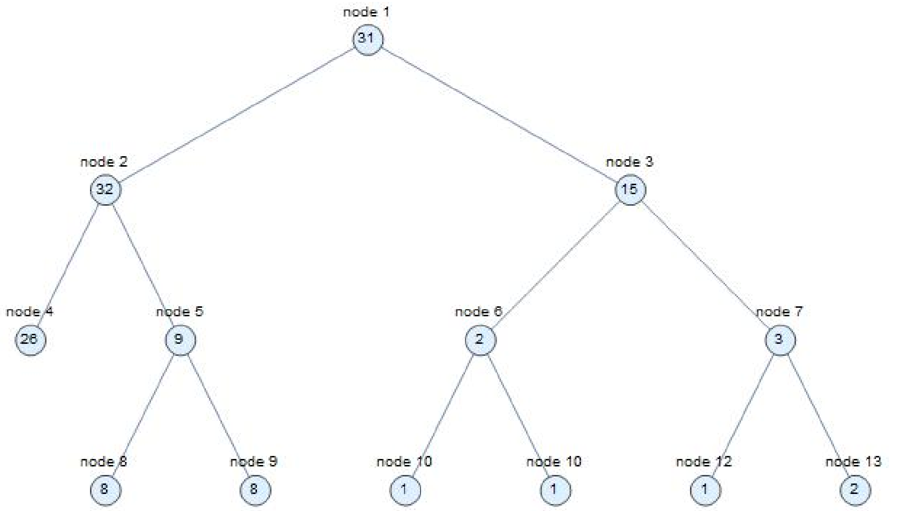
c) ○ Node 15 is the root, and its value is 1.

d) ○ Node 16 is the root, and its value is 3.

Question 2

Your answer is CORRECT.

A full binary tree is said to have the *heap* or *max heap* property if the value stored in a parent node is *always* bigger than or equal to the values stored in *any* of its children. We will use the notation $v(p)$ to denote the value stored at node p, which means that a full binary tree T has property *heap* if, whenever node c is a child of node p, it follows that $v(p) \geq v(c)$.



Does the full binary tree above have the heap property?

a) ○ Yes! This tree has property heap! (Heap, Heap Hurray!)

b) ● No! This tree does not have property heap. :(

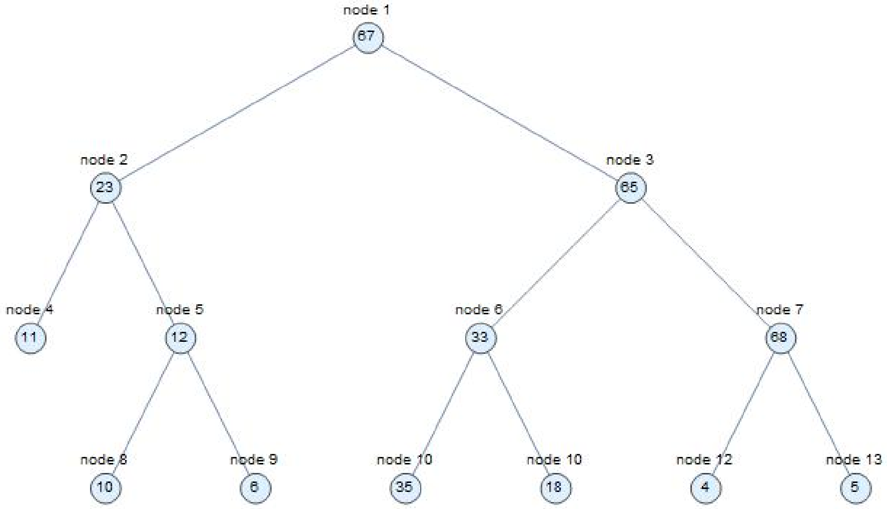
Question 3

Your answer is CORRECT.

The main point of this quiz question (and the next three) is to understand how "structural induction" can be used to prove the following proposition:

Proposition (The Max-Root Proposition). If a full binary tree, T, has the heap property, then $v(r) \geq v(n)$ for all nodes n.

Here, as usual, r denotes the root of T and $v(n)$ denotes the value (or "key") stored at node n. The full binary tree in the image below does *not* satisfy the heap property, and you can check that, in this particular example, the Proposition's conclusion is false.



Which, if any, of the following best describes a base case in our Structural Induction proof of the Max-Root Proposition?

a) ○ We would suppose a full binary heap-property tree T has nodes and then conclude that all the nodes have the same value.

b) ○ We would suppose T is a full binary tree with the heap property *and* then note that $v(r) \geq v(n)$ for all other nodes n.

c) ○ We would suppose T is a full binary tree with the heap property and satisfies $v(r) \geq v(n)$ for all nodes n.

d) ○ We would let T be a tree consisting of one node, r (and note that T is full binary *and* has the heap property for any stored value $v(r)$). Next we would check that $v(r) \geq v(n)$ for some node n.

e) ● We would let T be a tree consisting of one node, r (and note that T is full binary *and* has the heap property for any stored value $v(r)$). Next we would check that $v(r) \geq v(n)$ for all nodes n.

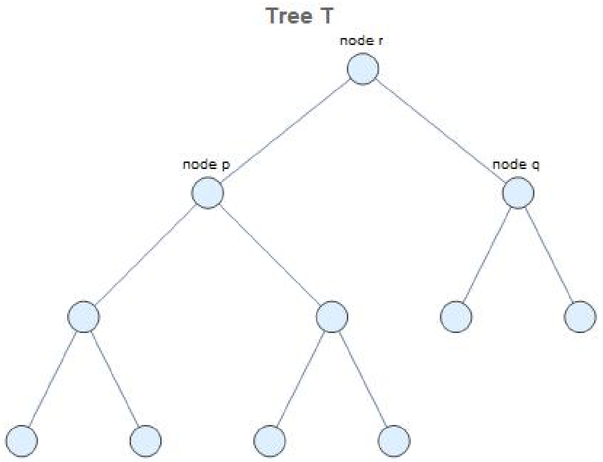
Question 4

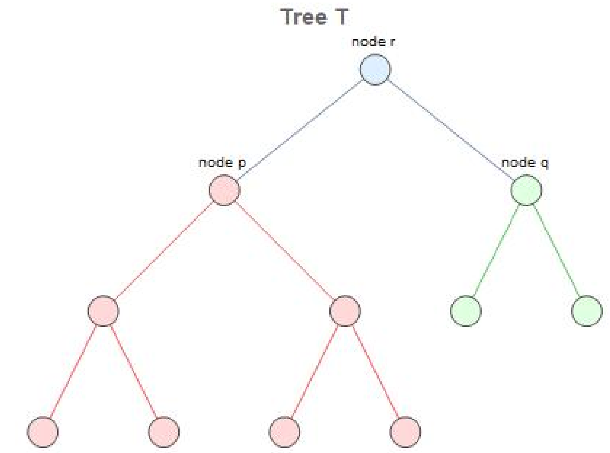
Your answer is CORRECT.

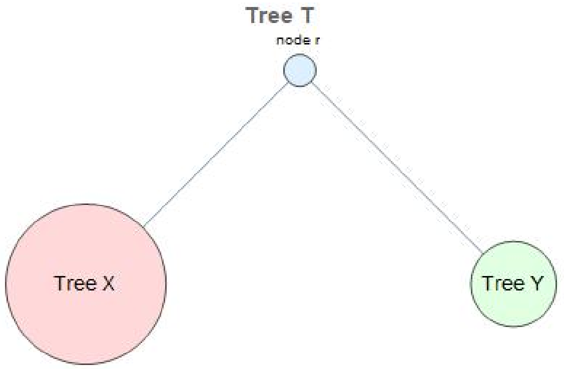
The main point of this quiz question (and the next two) is to continue our exploration of how "structural induction" can be used to prove the following proposition:

Proposition (The Max-Root Proposition). If a full binary tree, T, has the heap property, then $v(r) \geq v(n)$ for all nodes n.

Here, as usual, r denotes the root of T and $v(n)$ denotes the value (or "key") stored at node n. The image below shows a sample tree, T, that is full binary and is assumed to have the heap property; in fact, three images of T are displayed so as to clarify how to htink about any full binary tree as two subtrees -- X and Y -- joined to the root r.







Which, if any, of the following best describes the "structrual inductive" or "recursive" step for our proof of the Max-Root Proposition?

a) ○ We would suppose the Max-Root Proposition is true for some $k \in N$ and then show it is true for $k + 1$.

b) ● We would let T be a full binary, heap property tree that consists of a root r joining two subtrees X and Y (each of which have the heap-property). We would then assume the Max-Root Proposition is true for both X and Y and argue that $v(r) \geq v(n)$ for all nodes n.

c) ○ We would suppose the Max-Root Proposition is true for full binary trees with the heap property. This would then imply it is true for all full binary trees with the heap property.

d) ○ We would let T be a full binary, heap property tree that consists of a root r joining two subtrees X and Y (each of which have the heap-property). We would then assume the Max-Root Proposition is true for both X and Y and argue that $v(r) \leq v(n)$ for all nodes n.

e) ○ We would suppose a full binary heap-property tree T has nodes and then conclude that all the node's have the same value.

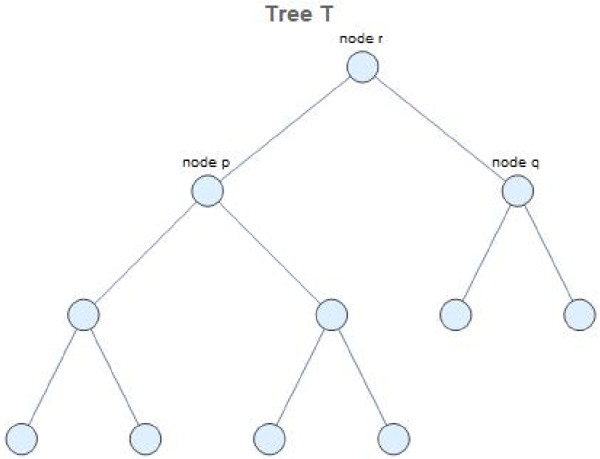
Question 5

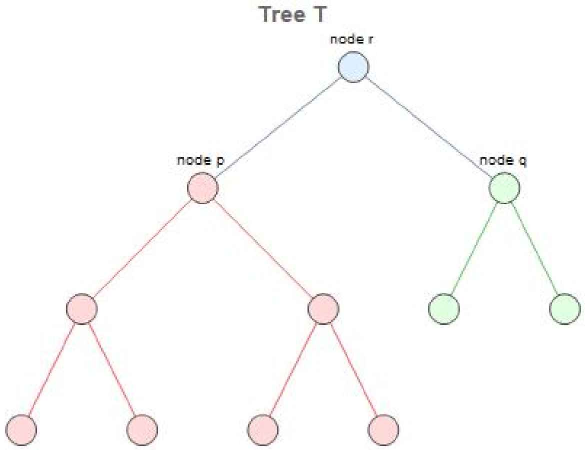
Your answer is CORRECT.

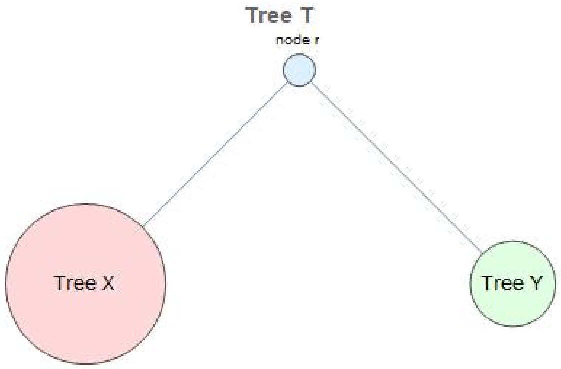
The main point of this quiz question (and the next one) is to continue our exploration of how "structural induction" can be used to prove the Max-Root Proposition:

Proposition (The Max-Root Proposition). If a full binary tree, T, has the heap property, then $v(r) \geq v(n)$ for all nodes n.

Here, as usual, r denotes the root of T and $v(n)$ denotes the value (or "key") stored at node n.







This question continues our work from the previous problem where, as part of our recursive step, we have assumed that a full binary heap-tree T consists of a root r joining two other full binary heap-trees, X and Y, each with respective roots p and q (see the images above for one example), and where the Max-Root Proposition is true for each. We now want to prove that $\mathrm{v(r)} \geq \mathrm{v(n)}$ for all nodes n. As a step in this direction, we need to understand why it is only necessary to prove that $\mathrm{v(r)} \geq \mathrm{v(p)}$ and $\mathrm{v(r)} \geq \mathrm{v(q)}$. Why do we need to only prove these two inequalities?

a) ● Since (by our inductive hypothesis) the Max-Root Proposition is true for X and Y, their highest values are each stored in the roots p and q.

b) ○ We only need to prove these two inequalities because X is a full binary tree with more than three nodes.

c) ○ Since (by our inductive hypothesis) the Max-Root Proposition is true for X and Y, their lowest values are each stored in the roots p and q.

d) ○ We only need to prove these inequalities because the picture does not show any values stored in nodes that are children of p and q.

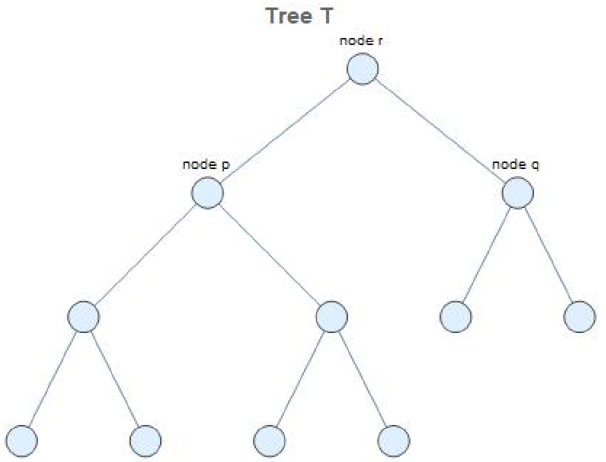
Question 6

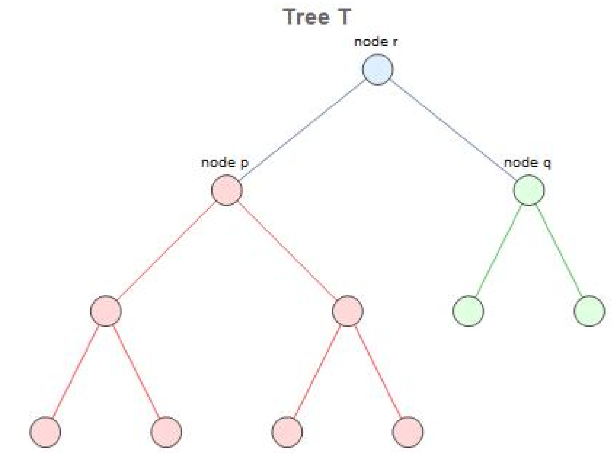
Your answer is CORRECT.

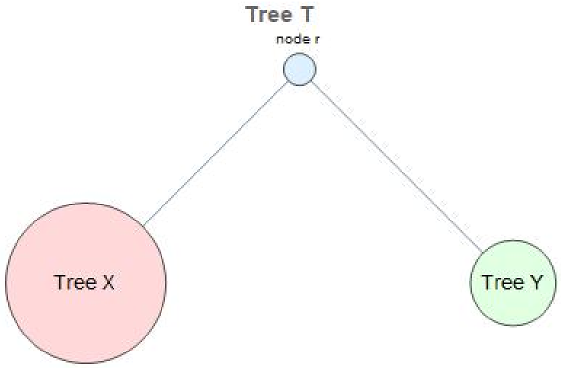
The main point of this quiz question is to continue our exploration of how "structural induction" can be used to prove the Max-Root Proposition:

Proposition (The Max-Root Proposition). If a full binary tree, T, has the heap property, then $\mathrm{v(r)} \geq \mathrm{v(n)}$ for all nodes n.

Here, as usual, r denotes the root of T and $\mathrm{v(n)}$ denotes the value (or "key") stored at node n.







This question continues our work from the previous two problems where, as part of our recursive step, we have assumed that a full binary heap-tree T consists of a root r joining two other full binary heap-trees, X and Y, each with respective roots p and q (see the images above for one example), and where the Max-Root Proposition is true for each.. To complete the proof we now need to answer one question: why is it true that $v(r) \geq v(p)$ and $v(r) \geq v(q)$?

a) ○ These inequalities hold because, by assumption, T has the heap property and r is the child of both p and q.

b) ○ These inequalities are not true (and so the Max-Root Proposition may be false)!

c) ● These inequalities hold because, by assumption, T has the heap property and p and q are children of r.

d) ○ These inequalities are true because if they were false then T would not be a binary tree.

e) ○ These inequalities hold because, by assumption, T has the heap property and p is a parent to q.

Question 7

Your answer is CORRECT.

Consider the set $S \subseteq N \times N$ recursively defined by the following rules:

$$(0, 0) \in S$$
$$(a, b) \in S \;\Rightarrow\; (a + 6, b + 5) \in S.$$

Which of the following is a true statement about S ?

a) ○ $S = R$.

b) ○ $|S| = 6$

c) ● $(0, 0)$ is a "base element" of S .

d) ○ $(24, 20)$ is a "base element" of S .

e) ○ $S = N \times N$

Question 8

Your answer is CORRECT.

Consider the set $S \subseteq N \times N$ recursively defined by the following rules:

$$(0,0) \in S$$

$$(a,b) \in S \Rightarrow (a+13, b+9) \in S.$$

Suppose we wish to prove a fact about the elements of S using Structural Induction (as we will happen in the next problem). From the options below, which could serve as the "base case" part of our proof?

a) ○ There is no base case part of a Structural Induction proof.

b) ○ Test the base case when $n = 0$.

c) ○ Demonstrate the fact is true on the "base elements" of S -- in this case there are 2 such base elements, $(13, 9)$ and $(14, 18)$.

d) ● Verify that the proposition is true for all of S 's "base elements."

e) ○ Verify that the proposition is true for the single element $(13, 27)$.

Question 9

Your answer is CORRECT.

Consider the set $S \subseteq N \times N$ recursively defined by the following rules:

$$(0,0) \in S$$

$$(a,b) \in S \Rightarrow (a+25, b+4) \in S.$$

Suppose we wish to use Structural Induction to prove the following proposition about the elements of S :

$$\text{Proposition. If } (x,y) \in S, \text{ then } x+y \text{ is a multiple of } 29.$$

Which of the following most accurately describes the "inductive" step of our proof?

a) ○ Suppose that the proposition is true for $(a, b) \in S$ and then prove it is true for the next element of S .

b) ○ Find a contradiction.

c) ○ Prove that, for any $(a, b) \in S$, $(a+b)|29 \Rightarrow (a+25+b+4)|29$.

d) ○ Prove That if $a+b$ is a multiple of 29, then $(a+1)+(b+1)$ is also a multiple of 29.

e) ● Prove that, for any $(a, b) \in S$, $29|(a+b) \Rightarrow 29|(a+25+b+4)$.

Question 10

Your answer is CORRECT.

Consider the set $S \subseteq N \times N$ recursively defined by the following rules:

$$(0,0) \in S$$

$$(a,b) \in S \Rightarrow (a+9, b+2) \in S.$$

Is the following proposition true?

$$\text{Proposition. If } (x,y) \in S, \text{ then } x+y \text{ is a multiple of } 11.$$

a) ● Yes! Structural Induction can be used to show this is true.

b) ○ No, Structrual Induction does not apply to this Proposition.