CS 173, Fall 2015
Examlet 9, Part A

NETID:

FIRST:	**LAST:**

Discussion: Thursday 2 3 4 5 Friday 9 10 11 12 1 2

(18 points) If T is a binary tree with root R, then Mass(T) is defined to be

- 0 if R is a leaf
- m if R has one child subtree, with Mass m
- 1+m if R has two child subtrees, both with Mass m
- otherwise, the maximum Mass of R's two child subtrees.

Use (strong) induction to prove that a binary tree T with Mass(T)=p has at least 2^p leaves

Solution: The induction variable is named **h** and it is the **height** of/in the tree.

Base Case(s): $h = 0$. In this case, the tree has only one node, so its Mass is 0. And it has $2^0 = 1$ leaves. So the claim is true for h=0.

Inductive Hypothesis [Be specific, don't just refer to "the claim"]: Any binary tree T with Mass(T)=p has at least 2^p leaves, for trees of height $h = 0, 1, \ldots, k-1$, $(k \geq 1)$.

Inductive Step: Let T be a tree of height k. There are three cases:

Case 1: T has one child subtree T_1 with Mass(T_1) = Mass(T) = q. By the inductive hypothesis T_1 has at least 2^q leaves, and T has the same number of leaves. So T also has at least 2^q leaves.

Case 2: T has two child subtrees T_L and T_R, where Mass(T_L)=Mass(T_M)=m and Mass(T)=m+1. By the inductive hypothesis T_L has at least 2^m leaves, and T_R has at least 2^m leaves. So T has at least 2^{m+1} leaves.

Case 3: T has two child subtrees T_L and T_R, where Mass(T_L)=m, Mass(T_M)=n, and Mass(T)=max(m,n). By the inductive hypothesis T_L has at least 2^m leaves, and T_R has at least 2^n leaves. So T has at least $2^{max(m,n)}$ leaves.

In all three cases, T has at least 2^q leaves, where q=Mass(T). This is what we needed to prove.

CS 173, Fall 2015
Examlet 9, Part A

NETID:

FIRST: | **LAST:**

Discussion: Thursday 2 3 4 5 Friday 9 10 11 12 1 2

(18 points) A Western tree is a full binary tree whose nodes contain integers such that

- Every leaf contains the value 0.
- The value $v(X)$ in a node X is (strictly) larger than the values in X's children.

Use (strong) induction to prove that the value in the root of a Western tree is larger than the value in any other node of the tree.

Solution: The induction variable is named **h** and it is the **height** of/in the tree.

Base Case(s): $h = 0$. A tree of height zero contains only one node. It's value is (vacuously) larger than all the other nodes in the tree because there are no other nodes.

Inductive Hypothesis [Be specific, don't just refer to "the claim"]: Suppose that the root contains the largest value for all Western trees of height $h = 0, 1, \ldots, k-1$

Inductive Step: Let T be a Western tree of height $k > 0$. Since T is a full binary tree, its root r has two children p and q. Suppose that X is the subtree rooted at p and Y is the subtree rooted at q.

Both X and Y have height $< k$. Moreover, notice that X and Y are Western trees because they are subtrees of T.

Suppose that x is any node of T, $x \neq r$. We need to show that $v(r) > v(x)$. There are three cases:

Case 1: x is the root of X or Y. Then $v(r) > v(x)$ by the definition of a Western tree.

Case 2: x is any node other than p in the subtree X. Then $v(p) > v(x)$ by the inductive hypothesis, and $v(r) > v(p)$ by the definition of a Western tree. So $v(r) > v(x)$.

Case 3: x is a node other than q in the subtree Y. Then $v(q) > v(x)$ by the inductive hypothesis, and $v(r) > v(q)$ by the definition of a Western tree. So $v(r) > v(x)$. [It's ok to say this is just like Case 2.]

So, for any node x in T, $v(r) > v(x)$.

CS 173, Fall 2015
Examlet 9, Part A

NETID:

FIRST: **LAST:**

Discussion: Thursday 2 3 4 5 Friday 9 10 11 12 1 2

(18 points) A Raven tree is a binary tree in which each node X contains an integer label $v(X)$ such that

- If X is a leaf, $v(X)$ is 7, 23, or 31.
- If X has one child Y, then $v(X) = v(Y) + 7$.
- If X has two children Y and Z, then $v(X) = v(Y)v(Z)$.

Use strong induction to prove that the value in the root of a Raven tree is always positive.

Solution: The induction variable is named **h** and it is the **height** of/in the tree.

Base Case(s): $h = 0$. The tree consists of a single leaf. $v(X)$ is 7, 23, or 31, all of which are positive.

Inductive Hypothesis [Be specific, don't just refer to "the claim"]: The value in the root of a Raven tree is always positive, for tree height $h = 0, 1, \ldots, k-1$ ($k \geq 1$).

Inductive Step: Let T be a Raven tree of height k, with root R. There are two cases:

Case 1: R has one child Y. By the inductive hypothesis, $v(Y)$ is positive. So $v(R) = v(Y) + 7$ must be positive.

Case 2: R has two children Y and Z. By the inductive hypothesis, $v(Y)$ and $v(Z)$ are positive. So $v(R) = v(Y)v(Z)$ must also be positive.

In both cases $v(R)$ is positive, which is what we needed to prove.

CS 173, Fall 2015
Examlet 9, Part A

NETID:

FIRST: **LAST:**

Discussion: **Thursday 2 3 4 5 Friday 9 10 11 12 1 2**

(18 points) Monster trees are binary trees whose nodes are labelled with strings, such that

- Each leaf node has label `left`, `right`, or `back`
- If a node has one child, it has label α`x` where where α is the child's label. E.g. if the child has label `left` then the parent has `leftx`.
- If a node has two children, it contains $\alpha\beta$ where α and β are the child labels. E.g. if the children have labels `right` and `back`, then the parent has label `rightback`.

Let S(n) be the length of the label on node n. Let L(n) be the number of leaves in the subtree rooted at n. Use (strong) induction to prove that $S(n) \geq 4L(n)$ if n is the root node of any Monster tree.

Solution: The induction variable is named **h** and it is the **height** of/in the tree.

Base case(s): $h = 0$. The tree consists of a single leaf node, so $L(n) = 1$. The node has label `left`, `right`, or `back`, so $S(n) \geq 4$. So $S(n) \geq 4L(n)$.

Inductive hypothesis [Be specific, don't just refer to "the claim"]:

Suppose that $S(n) \geq 4L(n)$ if n is the root node of any Monster tree of height $< k$ (where $k \geq 1$).

Rest of the inductive step:

Suppose that T is a Monster tree of height k. There are two cases:

Case 1: The root n of T has a single child node p. By the inductive hypothesis $S(p) \geq 4L(p)$. $L(n) = L(p)$. And $S(n) = S(p) + 1$. So $S(n) \geq 4L(n)$.

Case 2: The root n of T has two children p and q. By the inductive hypothesis $S(p) \geq 4L(p)$ and $S(q) \geq 4L(q)$.

Notice that $L(n) = L(p) + L(q)$. And $S(n) = S(p) + S(q)$.

So $S(n) = S(p) + S(q) \geq 4L(p) + 4L(q) = 4L(n)$.

CS 173, Fall 2015
Examlet 9, Part A

NETID:

FIRST: **LAST:**

Discussion: Thursday 2 3 4 5 Friday 9 10 11 12 1 2

(18 points) A Tippy tree is a full binary tree in which each node is colored orange or blue, such that:

- If v is a leaf node, then v is colored orange.
- If v has two children of the same color, then v is colored blue.
- If v has two children of different colors, then v is colored orange.

Use (strong) induction to show that the root of an Tippy tree is blue if and only if the tree has an even number of leaves. You may assume basic divisibility facts e.g. the sum of two odd numbers is even.

Solution: The induction variable is named **h** and it is the **height** of/in the tree.

Base Case(s): $h = 0$. The tree consists of a single node, which must be colored orange. The claim holds because the tree has an odd number of leaves (i.e. just one).

Inductive Hypothesis [Be specific, don't just refer to "the claim"]: Suppose that the root of an Tippy tree is blue if and only if the tree has an even number of leaves, for trees of height $h = 0, 1, \ldots, k-1$ (k an integer ≥ 1).

Inductive Step: Let T be a Tippy tree of height k. Since T is a full binary tree with height ≥ 1, T consists of a root plus two child subtrees. There are three cases:

Case 1: The root of T is blue and the roots of the child subtrees are both orange. By the inductive hypothesis, both subtrees have an odd number of leaves. Therefore T has an even number of leaves.

Case 2: The root of T is blue and the roots of the child subtrees are both blue. By the inductive hypothesis, both subtrees have an even number of leaves. Therefore T has an even number of leaves.

Case 3: The root of T is orange, the root of one child subtree (call it T_1) is orange, and the root of the old child subtree (call it T_2) is blue. By the inductive hypothesis, T_1 has an odd number of leaves and T_2 has an even number of leaves. Therefore T has an odd number of leaves.

In all three cases, the root of T is blue if and only if T has an even number of leaves, which is what we needed to prove.

CS 173, Fall 2015
Examlet 9, Part A

NETID:

FIRST: **LAST:**

Discussion: Thursday 2 3 4 5 Friday 9 10 11 12 1 2

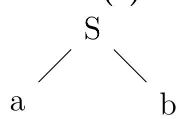
(18 points) Here is a grammar G, with start symbol S and terminal symbols a and b.

$$S \;\rightarrow\; a\,S\,b \;|\; b\,S\,a \;|\; S\,S \;|\; a\,b$$

Use (strong) induction to prove that any tree matching (aka generated by) grammar G has equal numbers of a's and b's. Use $A(T)$ and $B(T)$ as shorthand for the number of a's and b's in a tree T.

Solution: The induction variable is named **h** and it is the **height** of/in the tree.

Base Case(s): At $h = 1$, there is only one tree matching grammar G, which looks like



This contains the same number of a's and b's.

Inductive Hypothesis [Be specific, don't just refer to "the claim"]: Suppose that any tree matching (aka generated by) grammar G has equal numbers of a's and b's, for trees with heights $h = 1, 2, \ldots, k-1$ ($k \geq 1$).

Inductive Step: Let T be a tree of height k matching grammar G. Let r be the root of T. Since $k \geq 1$, there are three cases

Case 1: The children of r have labels (left to right): a, S, b. The middle child is the root of a subtree T_1. By the inductive hypothesis, T_1 contains equal numbers of a's and b's. $A(T) = A(T_1) + 1$ and $B(T) = B(T_1) + 1$. So $A(T) = B(T)$.

Case 2: The children of r have labels (left to right): b, S, a. The middle child is the root of a subtree T_1. This is exactly like Case 1.

Case 3: r has two child subtrees T_1 and T_2, both with a root labelled S. By the inductive hypothesis, $A(T_1) = B(T_1)$ and $A(T_2) = B(T_2)$. But all the a's and b's in T must live in the two subtrees, so $A(T) = A(T_1) + A(T_2)$ and $B(T) = B(T_1) + B(T_2)$. Combining these equations, we get that $A(T) = B(T)$.

In both cases $A(T) = B(T)$, which is what we needed to show.