Name:____

NetID:____________________ **Lecture: A B**

Discussion: Thursday Friday 10 11 12 1 2 3 4 5 6

(18 points) Sarah needs to saw a m by n by p inch block of wood into one-inch cubes. (m, n, and p are integers.) The saw can slice a block of wood at any integer position parallel to one of its sides. However, a safety feature prevents her from slicing more than one piece of wood at a time. Use (strong) induction to prove that it takes $mnp - 1$ cuts to divide the block of wood into one-inch cubes, for any sequence of cuts.

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

Base Case(s): At h=1, the block already consists of a single one-inch cube. So we don't need to divide it further. That is, we need 0 = h-1 cuts to divide it up. So the claim is true.

Inductive Hypothesis [Be specific, don't just refer to "the claim"]: Any block of volume h with integer side lengths can be divided into one-inch cubes using h-1 cuts, for h from 1 up through k-1.

Inductive Step: Suppose that B is a block of wood with integer side lengths and volume k, where $k > 1$. Let's cut B at any integer position parallel to one of the sides. This creates two blocks X and Y with integer side lengths. Let v_X be the volume of X and v_Y be the volume of Y. Then $k = v_X + v_Y$.

By the inductive hypothesis, we can reduce X to one-inch cubes using $v_X - 1$ cuts. Similarly, we can reduce Y to one-inch cubes using $v_Y - 1$ cuts.

Therefore, to reduce B to one-inch cubes, we use our initial cut, then divide X and Y using $v_X - 1$ and $v_Y - 1$ cuts (respectively). Then the total number of cuts required to divide up B is $1 + (v_X - 1) + (v_Y - 1) = v_X + v_Y - 1 = k - 1$. So dividing B into one-inch cubes requires $k - 1$ cuts, which is what we needed to prove.

Name:____

NetID:____________________ **Lecture:** **A** **B**

Discussion: **Thursday** **Friday** **10** **11** **12** **1** **2** **3** **4** **5** **6**

(18 points) Recall that a node in a full binary tree is either a leaf or has exactly two children. A Monkey tree is a full binary tree such the two child subtrees of each internal node have heights that differ by at most one. Prove that every Monkey tree of height h contains at least F_{h+1} leaves, where F_k is the kth Fibonacci number. (Recall: $F_0 = 0$, $F_1 = F_2 = 1$)

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

Base Case(s): At $h = 0$, the tree contains exactly one node and therefore exactly one leaf. Also, $F_{h+1} = F_1 = 1$, so the claim holds.

At $h = 1$, the tree must consist of three nodes: a root and its two children. So it has two leaves. $F_{h+1} = F_2 = 1$. So the number of leaves is $\geq F_{h+1}$.

Inductive Hypothesis [Be specific, don't just refer to "the claim"]: Suppose that every Monkey tree of height h contains at least F_{h+1} nodes, for $h = 0, 1, \ldots, k-1$.

Inductive Step: Let T be a Monkey tree of height k ($k \geq 1$). The root of T must have two child subtrees T_a and T_b, whose heights differ by at most one.

Case 1: T_a has height $k-1$ and T_b has height $k-2$. By the inductive hypothesis, T_a has at least F_k leaves and T_b has at least F_{k-1} leaves. So T must have at least $F_k + F_{k-1} = F_{k+1}$ leaves.

Case 2: T_a has height $k-2$ and T_b has height $k-1$. By the inductive hypothesis, T_a has at least F_{k-1} leaves and T_b has at least F_k leaves. So T must have at least $F_k + F_{k-1} = F_{k+1}$ leaves.

Case 3: T_a and T_b have height $k-1$. By the inductive hypothesis, T_a and T_b each have at least F_k leaves. So T must have at least $2F_k \geq F_k + F_{k-1} = F_{k+1}$ leaves.

In all cases, T must have at least F_{k+1} leaves, which is what we needed to prove.

Name:____

NetID:____________________ **Lecture:** **A** **B**

Discussion: **Thursday** **Friday** **10** **11** **12** **1** **2** **3** **4** **5** **6**

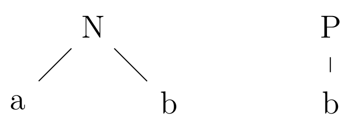
(18 points) Here is a grammar G, with start symbols N and P, and terminal symbols a and b.

$$\begin{aligned} N &\rightarrow P\ P \mid a\ b \\ P &\rightarrow N\ P \mid b \end{aligned}$$

Use (strong) induction to prove that any tree matching (aka generated by) grammar G has an even number of leaves if and only if its root has label N.

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

Base Case(s): The shortest trees matching grammar G have height $h = 1$. There are two such trees, which look like



The tree with root N has an even number of leaves and the tree with root P has an odd number of leaves, so the claim holds.

Inductive Hypothesis [Be specific, don't just refer to "the claim"]: Suppose that all trees T matching grammar G with heights $h = 1, 2, \ldots, k-1$ have an even number of leaves if and only if the root of T has label N, for some integer $k \geq 2$.

Inductive Step: Let T be a tree of height k matching grammar G, where $k \geq 2$. There are two cases:

Case 1: T consists of a root with label P, with a left child T_1 with root label N and a right child T_2 with root label P. By the inductive hypothesis, T_1 has an even number of leaves and T_2 has an odd number of leaves. Since the leaves in T are exactly the leaves in T_1 plus the leaves in T_2, T has an odd number of leaves.

Case 2: T consists of a root with label N, with child subtrees T_1 and T_2 that have root labels P. By the inductive hypothesis, T_1 and T_2 both have an even number of leaves, so T must have an even number of leaves.

In both cases, T has an even number of leaves if and only if the root of T has label N, which is what we needed to show.

Name:____

NetID:____________________ Lecture: A B

Discussion: Thursday Friday 10 11 12 1 2 3 4 5 6

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

- Every leaf node contains 7, 9, or 12.
- A node with one child contains the same number as its child.
- A node with two children contains the value $xy - y$, where x and y are the values in its children.

Use (strong) induction to prove that the value in the root of a Camel tree is always ≥ 7

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

Base Case(s): $h = 0$. Camel trees of height zero have a single node, which is both the root and a leaf. So it contains 7, 9, or 12, all of which are ≥ 7.

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

Suppose that the value in the root node of a Camel tree is always ≥ 7, for all trees of height $h = 0, 1, \ldots, k-1$ (k an integer ≥ 1).

Inductive Step: Let T be a Camel tree of height k ($k \geq 1$). There are two cases:

Case 1: T consists of a root with a single subtree under it. Call the subtree T_1. By the inductive hypothesis, the root of T_1 contains a value ≥ 7. By the definition of Camel trees, the root of T contains the same value, which is therefore also ≥ 7.

Case 2: T consists of a root with two subtrees T_1 and T_2 under it. Suppose the roots of the subtrees contain values x and y. By the inductive hypothesis $x \geq 7$ and $y \geq 7$.

The root of T then has value $xy - y$ by the definition of Camel trees. Since $x \geq 7$, $x - 1 \geq 6 \geq 1$. So $xy - y = (x-1)y \geq y \geq 7$.

In both cases the root of T contains a value ≥ 7, which is what we needed to show.

Name:____

NetID:________________________ **Lecture:** **A** **B**

Discussion: **Thursday** **Friday** **10** **11** **12** **1** **2** **3** **4** **5** **6**

(18 points) UIUC is considering hosting massive hackathons in rooms like the first floor of the Armory. Facilities will need to divide this $100h$ square foot room into h workspaces, each 100 square feet, using expanding partitions. Each end of each partition must be attached to a wall of the room or to another partition. A partition can expand to any length but cannot cross another partition. The partitions are low enough that doors are not required. Use (strong) induction to prove that they will need $h-1$ partitions, no matter how they arrange the partitions.

Solution: The induction variable is named __**h**__ and it is the __**area/100**__ of/in the room.

Base Case(s): For h=1, we need to divide the room into one workspace. But that's already the case, so we don't need any partitions. Since $h-1=0$, the claim holds.

Inductive Hypothesis [Be specific, don't just refer to "the claim"]: Suppose that $h-1$ partitions are required to divide a $100h$ square foot room, for $h = 1, \ldots, k-1$.

Inductive Step: Suppose we want to divide a $100k$ square foot room, where $k \geq 2$. Suppose Facilities uses their first partition to divide the room into two smaller rooms. Let's call them R_1 and R_2. Each smaller room must be a multiple of 100 square feet (otherwise we have no hope of making the right number of workspaces). Suppose Facilities has positioned the divider so that R_1 is $100p$ square feet. Then R_2 must be $100(k-p)$ square feet.

By the inductive hypothesis, they will require $p-1$ partitions to completely subdivide R_1 into workspaces, and $(k-p)-1$ partitions to completely subdivide R_2. Adding up these numbers, plus the first partition, we have a total of $(p-1)+(k-p-1)+1 = k-1$ partitions.

So dividing a room with area $100k$ square feet requires $k-1$ partitions, which is what we needed to prove.

Name:____

NetID:____________________ **Lecture:** **A** **B**

Discussion: **Thursday** **Friday** **10** **11** **12** **1** **2** **3** **4** **5** **6**

(18 points) Octopus 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 Octopus 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 Octopus tree of height $< k$ (where $k \geq 1$).

Rest of the inductive step:

Suppose that T is a Octopus 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 $S(n) = S(p) + S(q) \geq 4L(p) + 4L(q) = 4L(n)$.