CS 173, Spring 19 Examlet 9, Part A

Name:____

NetID:____________________ **Lecture: A B**

Discussion: Thursday Friday 9 10 11 12 1 2 3 4 5 6

(18 points) A Mouse tree is a binary tree containing 2D points such that:

- Each leaf node contains $(3, 1)$, $(-2, -5)$, or $(2, 2)$.
- An internal node with one child labelled (a, b) has label $(a + 1, b - 1)$.
- An internal node with two childen labelled (x, y) and (a, b) has label $(x + a, y + b)$.

Use (strong) induction to prove that the point in the root node of any Mouse tree is on or below the line $x = y$.

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

Base Case(s): The shortest Mouse trees consist of a single node containing $(3, 1)$, $(-2, -5)$, or $(2, 2)$. All three of these points lie on or below the line $x = y$.

Inductive Hypothesis [Be specific, don't just refer to "the claim"]: Suppose that the point in the root node of any Mouse tree is on or below the line $x = y$, for trees of height $h = 0, 1, \ldots, k - 1$. $(k \geq 1)$.

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

Case 1: The root of T has one child subtree, whose root contains (a, b). By the inductive hypothesis, (a, b) is on or below $x = y$, i.e. $b \leq a$. By the definition of a Mouse tree, the root of T contains $(a+1, b-1)$. Since $b \leq a$, $b - 1 \leq a + 1$, so this point is on or below $x = y$.

Case 2: The root of T has two child subtrees, whose roots contain (x, y) and (a, b). Then the root of T contains $(x + a, y + b)$. By the inductive hypothesis, $y \leq x$ and $b \leq a$, so $y + b \leq x + a$. So $(x + a, y + b)$ is on or below $x = y$.

In both cases the root note contains a point on or below $x = y$, which is what we needed to show.

Name:____

NetID:____________________ **Lecture: A B**

Discussion: Thursday Friday 9 10 11 12 1 2 3 4 5 6

(18 points) A palindrome is a string that is the same if you reverse it. For example, `abbabba` and `abaaba` are palindromes. The empty (zero-length) string ϵ counts as a palindrome.

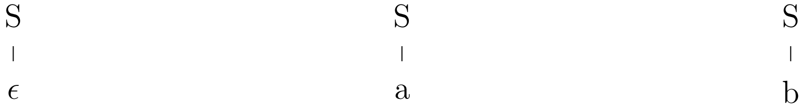
Here is a grammar G, with start symbol S and terminal symbols a and b.

$$S \rightarrow a\ S\ a\ \mid\ b\ S\ b\ \mid\ a\ \mid\ b \mid\ \epsilon$$

Use (strong) induction to prove that any palindrome made out of characters a and b can be generated by grammar G. That is, show how to build parse trees for these strings. Hint: remove the first and last character from the string.

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

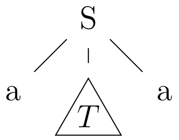
Base Case(s): At $h = 0$, the only palindrome is the empty string ϵ. At $h = 1$, the only palindromes are `a` and `b`. These can be generated by G as follows:



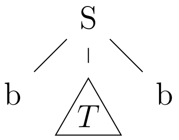
Inductive Hypothesis [Be specific, don't just refer to "the claim"]: Suppose that any palindrome made out of a's and b's can be generated by grammar G, for strings of length $h = 0, 1, \ldots, k-1$ ($k \geq 2$).

Inductive Step: Let w be a palindrome of length k. Since $k \geq 2$, there are two cases

Case 1: w starts with an a. Since w is a palindrome, it must look like $w = ava$ where v is a string of length $k-2$. By the inductive hypothesis, we can build a parse tree T for v. We can then build a parse tree for w using the rule $S \rightarrow a\ S\ a$ like this



Case 2: w starts with a b. Since w is a palindrome, it must look like $w = bvb$ where v is a string of length $k-2$. By the inductive hypothesis, we can build a parse tree T for v. We can then build a parse tree for w using the rule $S \rightarrow b\ S\ b$ like this



In both cases, we can build a parse tree for the string w, which is what we needed to show.

Name:____

NetID:______________________ **Lecture:** **A** **B**

Discussion: **Thursday** **Friday** **9** **10** **11** **12** **1** **2** **3** **4** **5** **6**

(18 points) Recall that a node in a full binary tree must have either 0 or 2 children. A Shark 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 a Shark 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 a Shark 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 Shark 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.

Name:____

NetID:____________________ **Lecture: A B**

Discussion: Thursday Friday 9 10 11 12 1 2 3 4 5 6

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

- Every leaf node contains 5, 17, or 23.
- A node with one child contains the same number as its child.
- A node with two children contains the value $x(y+1)$, where x and y are the values in its children.

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

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

Base Case(s): The smallest Horse trees consist of a single root node, which is also a leaf. By the definition of Horse tree, this must contain 5, 17, or 23, all of which are positive.

Inductive Hypothesis [Be specific, don't just refer to "the claim"]: Suppose that the root of a Horse tree of height h is always positive, for $h = 0, \ldots, k-1$.

Inductive Step: Let T be a Horse tree of height k. There are two cases for what the top of T looks like.

Case 1: T consists of a root r with a single subtree S under it. r contains the same number as the root of S. Since S must be shorter than k, its root contains a positive number by the inductive hypothesis. Since r has the same label, r contains a positive number.

Case 2: T consists of a root r with a two subtrees S_1 and S_2. Suppose that the roots of S_1 and S_2 contain the numbers x and y. Then, by the definition of Horse tree, r contains $x(y+1)$.

Since S_1 and S_2 are shorter than k, x and y must be positive by the inductive hypothesis. Since y is positive, so is $y+1$. Since x and $y+1$ are positive, so is $x(y+1)$. So the root of T contains a positive number.

So, in either case, the root of T contains a positive number.

CS 173, Spring 19 Examlet 9, Part A

Name:____

NetID:____________________ **Lecture: A B**

Discussion: Thursday Friday 9 10 11 12 1 2 3 4 5 6

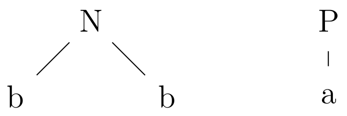
(18 points) Here is a grammar G, with start symbols N and P, and terminal symbols a and b.

$$\begin{array}{lcl} N & \rightarrow & P\,a \mid b\,b \\ P & \rightarrow & P\,N \mid a \end{array}$$

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 P and a right child T_2 with root label N. By the inductive hypothesis, T_1 has an odd number of leaves and T_2 has an even 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, whose children are a subtree T_1 on the left and a leaf node on the right. T_1 has root label P and the leaf has label a. By the inductive hypothesis, T_1 has an odd 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 9 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 Possum tree is a full binary tree whose leaves are all orange and whose root is blue.

Use (strong) induction to prove that a Possum tree contains a blue node with (at least) one orange child.

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

Base Case(s): A Possum tree must have height at least 1 because the root is a different color from the leaves. At $h = 1$, the tree consists of a blue root with two orange children, so the claim is true.

Inductive Hypothesis [Be specific, don't just refer to "the claim"]: Suppose that every Possum tree contains a blue node with (at least) one orange child, for heights $h = 0, \ldots, k-1$.

Inductive Step: Consider a Possum tree T with height k. T consists of a root node x (colored blue) and two child subtrees T_L and T_R.

There are three cases:

Case 1: T_L has an orange root. Then x is the required blue node with at least one orange child.

Case 2: T_R has an orange root. Then x is the required blue node with at least one orange child.

Case 3: T_L and T_R both have blue roots. Then we can apply the inductive hypothesis to T_L. By the inductive hypothesis, T_L must contain a blue node with at least one orange child. Since T_L is a subtree of T, this node and its children also live in T.

In all three cases, T contains a blue node with at least one orange child, which is what we needed to prove.

CS 173, Spring 19 Examlet 9, Part A

Name:____

NetID:______________________ **Lecture: A B**

Discussion: Thursday Friday 9 10 11 12 1 2 3 4 5 6

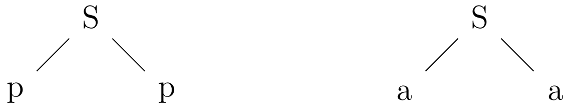
(18 points) Here is a grammar G, with start symbol S and terminal symbols a and p.

$$S \rightarrow S\,S \mid p\,S\,p \mid p\,p \mid a\,a$$

Use (strong) induction to prove that any tree matching (aka generated by) grammar G has an even number of nodes with label p. Use $P(T)$ as shorthand for the number of p's in a tree T.

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



Both of these contain an even number of nodes with label p.

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 $P(T)$ even, 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 S plus two child subtrees T_1 and T_2. By the inductive hypothesis $P(T_1)$ and $P(T_2)$ are both even. But $P(T) = P(T_1) + P(T_2)$. So $P(T)$ is also even.

Case 2: T consists of a root with label S plus three children. The left and right children are single nodes containing label p. The center child is a subtree T_1. By the inductive hypothesis, $P(T_1)$ is even. $P(T) = P(T_1) + 2$. So $P(T)$ is also even.

In both cases $P(T)$ is even, which is what we needed to show.

CS 173, Spring 19 Examlet 9, Part A

Name:____

NetID:________________________ **Lecture: A B**

Discussion: Thursday Friday 9 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 Snake tree is a full binary tree whose leaves are all blue and whose root is orange.

Use (strong) induction to prove that a Snake tree contains an orange node with two blue children.

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

Base Case(s): A Snake tree must have height at least 1 because the root is a different color from the leaves. At $h = 1$, the tree consists of an orange root with two blue children, so the claim is true.

Inductive Hypothesis [Be specific, don't just refer to "the claim"]: Suppose that every Snake tree contains an orange node with two blue children, for heights $h = 0, \ldots, k-1$.

Inductive Step: Consider a Snake tree T with height k. T consists of a root node x (colored orange) and two child subtrees T_L and T_R.

There are three cases:

Case 1: Both T_L and T_R have blue roots. Then x is the required orange node with two blue children.

Case 2: T_L has an orange root. Then we can apply the inductive hypothesis to T_L. By the inductive hypothesis, T_L must contain an orange node with two blue children. Since T_L is a subtree of T, this node and its children also live in T.

Case 3: T_R has an orange root. Then we can apply the inductive hypothesis to T_R. By the inductive hypothesis, T_R must contain an orange node with two blue children. Since T_R is a subtree of T, this node and its children also live in T.

In all three cases, T contains an orange node with two blue children, which is what we needed to prove.

CS 173, Fall 18 Examlet 9, Part A

Name:____

NetID:________________________ Lecture: B

Discussion: Friday 11 12 1 2 3 4

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

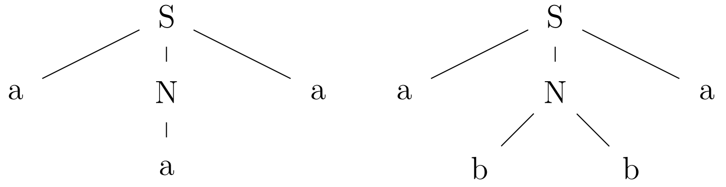
$$\begin{aligned} S &\rightarrow a\,S\,a \mid S\,a\,S \mid a\,N\,a \\ N &\rightarrow a \mid b\,b \end{aligned}$$

Use (strong) induction to prove that any tree of height h matching (aka generated by) grammar G has at least h nodes with label a. Use $A(T)$ as shorthand for the number of a's in a tree T.

Solution:

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

Base Case(s): The shortest trees generated by G have $h = 2$. They are as shown below and, as you can see, they both have at least two nodes labelled a.



Inductive Hypothesis [Be specific, don't just refer to "the claim"]: All trees of height h generated by G have at least h nodes labelled a, for $h = 2, 3, \ldots, k-1$. $(k \geq 3)$

Inductive Step: Suppose that T is a tree generated by G of height k. There are two cases:

Case 1: T consists of a root labelled S, with three children. The left and right children have label a. The middle child is a subtree T_1 whose root has label S. T_1 must have height $k-1$ so, by the inductive hypothesis, it contains at least $k-1$ a's. So T contains at least $(k-1)+2 = k+1$ a's.

Case 1: T consists of a root labelled S, with three children. The middle child child has label a. The left and right children are subtrees T_1 and T_2 whose roots have label S. At least one of these two subtrees has height $k-1$ so, by the indutive hypothesis, it contains at least $k-1$ a's. The middle child of T adds another a, so T must have at least k a's. (The other subtree may add additional a's.)

In either case, T has at least k nodes labelled a, which is what we needed to show.

CS 173, Fall 18 Examlet 9, Part A

Name:____

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

Discussion: **Friday 11 12 1 2 3 4**

(18 points) Recall that a node in a full binary tree must have either 0 or 2 children. A Sleepy 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 may be colored orange or blue.
- 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 a Sleepy tree is blue if and only if the tree has an even number of orange 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): A Sleepy tree with $H = 0$ consists of a single node. If it's blue, the tree contains no orange nodes, which is even. If it's orange, the tree contains one orange node, which is odd. In both cases, the claim is true.

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

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

Case 1: the root of T is colored blue and it has two child subtrees whose roots are the same color. If both are blue, then both subtrees contain an even numbr of orange leaves by the inductive hypothesis. Similarly, if both are orange, then each contains an odd number of orange leaves. Since two odd numbers, or two even numbers, sum to an even number, T has an even number of orange leaves.

Case 2: the root of T is colored orange and it has two child subtrees whose roots are opposite colors. By the inductive hypothesis, the subtree with an orange root contain an odd number of orange leaves and the subtree with a blue root contains an even number of orange leaves. So T contains an odd number of orange leaves.

In both cases, T contains an even number of orange leaves if and only if its root is blue, which is what we needed to show.

CS 173, Spring 18 Examlet 9, Part A

Name:____

NetID:____________________ **Lecture: A B**

Discussion: Thursday Friday 9 10 11 12 1 2 3 4 5 6

(18 points) Recall that F_n is the nth Fibonacci number, and these start with $F_0 = 0$, $F_1 = 1$.

Let T_n be the number of bit strings of length n that don't contain any consecutive zeros. E.g. when counting strings of length 6, we include 010110, but not 101001. Prove that $T_n = F_{n+2}$ for any natural number n. Hint: if w is a string with no consecutive zeros, either $w = 1x$, where x is a shorter string, or $w = 01y$, where y is a shorter string.

Solution: The induction variable is named__**n**__and it is the __**length**__ of/in the string.

Base Case(s): At $n = 0$, we have only the empty string ϵ. So $T_0 = 1 = F_2$.

At $n = 1$, we have two strings 0 and 1. So $T_1 = 2 = F_3$.

Inductive Hypothesis [Be specific, don't just refer to "the claim"]: Prove that $T_n = F_{n+2}$ for $n = 0, 1, \ldots, k-1$.

Inductive Step: Now consider $n = k$. We need to calculate T_k which is the number of bit strings of length k with no consecutive zeros. Following the hint, these strings come in two mutually-exclusive types:

Type 1: strings of the form $1x$, where x is a string of length $k-1$. By the induction hypothesis, there are F_{k+1} choices for x. So there are F_{k+1} choices for $1x$.

Type 2: strings of the form $01y$, where y is a string of length $k-2$. By the induction hypothesis, there are F_k choices for y. So there are F_k choices for $1y$.

T_k is equal to the number of strings of type 1 plus the number of strings of type 2. So $T_k = F_{k+1} + F_k$. By the definition of Fibonacci numbers, this is equal to F_{k+2}. So $T_k = F_{k+2}$, which is what we needed to show.

CS 173, Spring 18 Examlet 9, Part B

Name:____

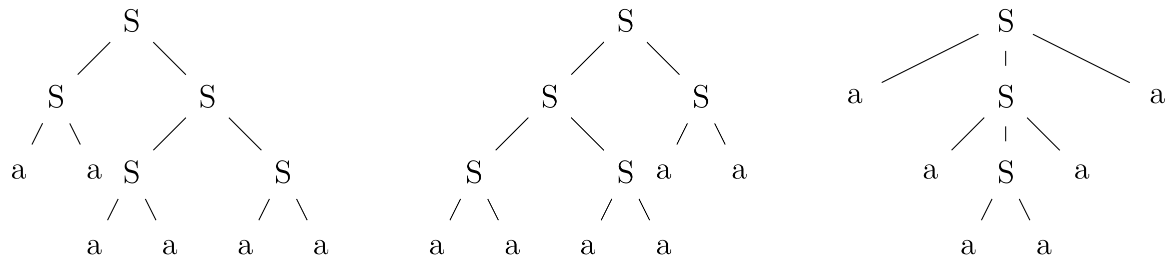
NetID:____________________ **Lecture:** **A** **B**

Discussion: **Thursday** **Friday** **9** **10** **11** **12** **1** **2** **3** **4** **5** **6**

1. (8 points) Here is a grammar with start symbol S and terminal symbol a. Draw three parse trees for the string `aaaaaa` that match this grammar.

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

Solution:



2. (4 points) Check the (single) box that best characterizes each item.

Total number of leaves in a 3-ary tree of height h 3^h □ $\leq 3^h$ ☑ $\frac{1}{2}(3^{h+1}-1)$ □ $3^{h+1}-1$ □

The level of a leaf node in a tree of height h. 0 □ 1 □ $h-1$ □ $\leq h$ ☑ h □

CS 173, Fall 17 Examlet 9, Part A

Name:____

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

Discussion: **Thursday** **Friday** **9** **10** **11** **12** **1** **2** **3** **4** **5** **6**

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

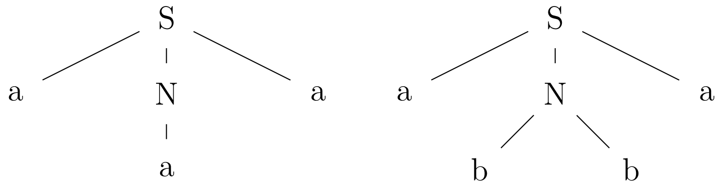
$$\begin{array}{lcl} S & \rightarrow & a\,S\,a \mid S\,a\,S \mid a\,N\,a \\ N & \rightarrow & a \mid b\,b \end{array}$$

Use (strong) induction to prove that any tree of height h matching (aka generated by) grammar G has at least h nodes with label a. Use $A(T)$ as shorthand for the number of a's in a tree T.

Solution:

The induction variable is named___**h**___and it is the ___**height**___ of/in the tree.

Base Case(s): The shortest trees generated by G have $h = 2$. They are as shown below and, as you can see, they both have at least two nodes labelled a.



Inductive Hypothesis [Be specific, don't just refer to "the claim"]: All trees of height h generated by G have at least h nodes labelled a, for $h = 2, 3, \ldots, k-1$. $(k \geq 3)$

Inductive Step: Suppose that T is a tree generated by G of height k. There are two cases:

Case 1: T consists of a root labelled S, with three children. The left and right children have label a. The middle child is a subtree T_1 whose root has label S. T_1 must have height $k-1$ so, by the inductive hypothesis, it contains at least $k-1$ a's. So T contains at least $(k-1)+2 = k+1$ a's.

Case 1: T consists of a root labelled S, with three children. The middle child child has label a. The left and right children are subtrees T_1 and T_2 whose roots have label S. At least one of these two subtrees has height $k-1$ so, by the indutive hypothesis, it contains at least $k-1$ a's. The middle child of T adds another a, so T must have at least k a's. (The other subtree may add additional a's.)

In either case, T has at least k nodes labelled a, which is what we needed to show.

Name:____

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

Discussion: **Thursday** **Friday** **9** **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 Lemon 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 Lemon tree of height h contains at least F_{h+1} nodes, 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 three nodes. $F_{h+1} = F_2 = 1$. So the number of leaves is $\geq F_{h+1}$.

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

Inductive Step: Let T be a Lemon tree of height k ($k \geq 2$). 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 nodes and T_b has at least F_{k-1} nodes. So T must have at least $F_k + F_{k-1} = F_{k+1}$ nodes.

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} nodes and T_b has at least F_k nodes. So T must have at least $F_k + F_{k-1} = F_{k+1}$ nodes.

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 nodes. So T must have at least $2F_k \geq F_k + F_{k-1} = F_{k+1}$ nodes.

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

Name:____

NetID:______________________ **Lecture: A B**

Discussion: Thursday Friday 9 10 11 12 1 2 3 4 5 6

(18 points) Suppose that G is a connected graph. A Friendly coloring of G labels each node orange or blue, such that

- If G contains only one node, it is colored orange, and
- Otherwise, every node of G is adjacent to at least one node of the opposite color.

Use (strong) induction to prove that any connected graph can be given a Friendly coloring. Hint: remove any node x (no special properties required) and color the rest of the graph. What color pattern is required if x is the only neighbor of another node?

Solution: The induction variable is named **h** and it is the **number of nodes** of/in the graph.

Base Case(s): $h = 1$. That is, the graph has a single node. Coloring it orange gives us a Friendly coloring.

Inductive Hypothesis [Be specific, don't just refer to "the claim"]: Any connected graph can be given a Friendly coloring, for graphs with number of nodes $h = 1, \ldots, k-1$, $(k \geq 2)$.

Inductive Step: Let G be a connected graph with k nodes $k \geq 2$. Pick one node x. Remove x (and all its edges) to create a smaller graph G'.

G' consists of some number of connected components $C_1, \ldots, C_p$. Each component has at most $k-1$ nodes, so it can be given a Friendly coloring by the inductive hypothesis. We need to assign a color to x to create a Friendly coloring for the full graph G.

Most nodes in G' have a neighbor of opposite color in the same component of G', and therefore they still satisfy this condition in G. So we just need to ensure that x gets a color different from one of its neighbors. Exception: if x is the only neighbor of a node y, then y was the only node in its component of G'. x must be colored blue to ensure that y (which was colored orange) has an opposite-color neighbor in G.

So there are three cases:

Case 1: All x's neighbors have the same color. Then we assign the opposite color to x.

Case 2: x is the only neighbor of a node y. Then we color x blue.

Case 3: Otherwise, we can assign either color to x.

This gives us a Friendly coloring of G, which is what we needed to construct.

Name:____

NetID:______________________ **Lecture: A B**

Discussion: Thursday Friday 9 10 11 12 1 2 3 4 5 6

(18 points) Recall that a node in a full binary tree must have either 0 or 2 children. A Happy 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 may be colored orange or blue.
- 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 a Happy tree is blue if and only if the tree has an even number of orange 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): A Happy tree with $H = 0$ consists of a single node. If it's blue, the tree contains no orange nodes, which is even. If it's orange, the tree contains one orange node, which is odd. In both cases, the claim is true.

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

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

Case 1: the root of T is colored blue and it has two child subtrees whose roots are the same color. If both are blue, then both subtrees contain an even numbr of orange leaves by the inductive hypothesis. Similarly, if both are orange, then each contains an odd number of orange leaves. Since two odd numbers, or two even numbers, sum to an even number, T has an even number of orange leaves.

Case 2: the root of T is colored orange and it has two child subtrees whose roots are opposite colors. By the inductive hypothesis, the subtree with an orange root contain an odd number of orange leaves and the subtree with a blue root contains an even number of orange leaves. So T contains an odd number of orange leaves.

In both cases, T contains an even number of orange leaves if and only if its root is blue, which is what we needed to show.

CS 173, Fall 17 Examlet 9, Part A

Name:______________________________________

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

Discussion: **Thursday** **Friday** **9** **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 Tidy tree is a full binary tree such that leaf nodes have label 0 and an internal node with label x matches one of these patterns:

- The left subtree is a leaf and the right subtree has root label y, where $y \equiv x - 1 \pmod 3$, or
- The roots of both subtrees have label y, where $y \equiv x - 1 \pmod 3$.

Use (strong) induction to prove that the root label of every Tidy tree is congruent to $h \pmod 3$, where h is the height of the tree. You may assume basic facts about congruence and modular arithmetic (e.g. congruence is transitive).

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, which is a leaf and therefore labelled 0. So the label is congruent to the height mod 3.

Inductive Hypothesis [Be specific, don't just refer to "the claim"]: Suppose that every Tidy tree of height h has a root label y which is congruent to $h \pmod 3$, for $h = 0, \ldots, k - 1$.

Inductive Step: Let T be a Tidy tree of height k ($k \geq 1$). Since $k \geq 1$, the root of T is an internal node. So there are two cases:

Case 1: The left subtree is a leaf and the right subtree has root label y, where $y \equiv x - 1 \pmod 3$. The right subtree must be at least as tall as the left subtree, so the right subtree has height $i - 1$. By the inductive hypothesis, we know that $y \equiv i - 1 \pmod 3$.

Case 2: The roots of both subtrees have label y, where $y \equiv x - 1 \pmod 3$. The taller subtree must have height $k - 1$. So, by the inductive hypothesis (applied to that subtree), $y \equiv h - 1 \pmod 3$.

So $y \equiv h - 1 \pmod 3$ in both cases. In both cases, we also have $y \equiv x - 1 \pmod 3$. So we must have $x - 1 \equiv k - 1 \pmod 3$. So $x \equiv k \pmod 3$, which is what we needed to prove.

Name:____

NetID:________________________ **Lecture: A B**

Discussion: Thursday Friday 9 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 Trim 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 Trim tree of height h contains at least $2^{h/2}$ leaves. You may use the fact that $\sqrt{2} > 1.4$.

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. $2^{h/2} = 2^0 = 1$, 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. $2^{h/2} = 2^{1/2} = \sqrt{2} < 2$, so the claim holds.

Inductive Hypothesis [Be specific, don't just refer to "the claim"]: Suppose that every Trim tree of height h contains at least $2^{h/2}$ nodes, for $h = 0, 1, \ldots, k-1$.

Inductive Step: Let T be a Trim tree of height k ($k \geq 2$). 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 $2^{(k-1)/2}$ leaves and T_b has at least $2^{(k-2)/2}$ leaves. So the number of leaves in T is at least

$$2^{(k-1)/2} + 2^{(k-2)/2} = (2^{-1/2} + 2^{-1})2^{k/2} = (\frac{1}{\sqrt{2}} + \frac{1}{2})2^{k/2} = \frac{\sqrt{2}+1}{2}2^{k/2} > \frac{1.4+1}{2}2^{k/2} > 2^{k/2}$$

Case 2: T_a has height $k-2$ and T_b has height $k-1$. This is exactly the same as case 1, except for swapping the roles of T_a and T_b.

Case 3: T_a and T_b have height $k-1$. By the inductive hypothesis, T_a and T_b each have at least $2(k-1)/2$ leaves. So the number of leaves in T must be at least $2 \cdot 2(k-1)/2 = 2^{(k+1)/2} > 2^{k/2}$.

In all cases, T must have at least $2^{k/2}$ 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) 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**

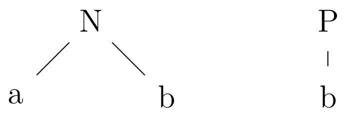
(18 points) Here is a grammar G, with start symbols N and P, and terminal symbols a and b.

$$\begin{array}{lcl} N & \to & P\ P \mid a\ b \\ P & \to & N\ P \mid b \end{array}$$

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