CSE 311 Section 07

Strong & Structural Induction

Announcements & Reminders

- Midterm
  - Good job!
  - Please don’t talk about the midterm!! Not everyone has taken it yet ☺

- HW6
  - Due Wednesday 5/15 @ 11:59pm

Strong Induction

Task 1 – Walk the Dawgs

Suppose that a dog walker takes care of $n \geqslant 12$ dogs. The dog walker is not a strong person, and will walk dogs in groups of 3 or 7 at a time (every dog gets walked exactly once). Prove that the dog walker can always split the n dogs into groups of 3 dogs or 7 dogs.

Task 1 – Walk the Dawgs

Suppose that a dog walker takes care of $n \geqslant 12$ dogs. The dog walker is not a strong person, and will walk dogs in groups of 3 or 7 at a time (every dog gets walked exactly once). Prove that the dog walker can always split the n dogs into groups of 3 dogs or 7 dogs.

Let $P(n)$ be "a group with n dogs can be split into groups of 3 dogs or 7 dogs." We will prove $P(n)$ for all natural numbers $n \geqslant 12$ by strong induction.

Task 1 – Walk the Dawgs

Suppose that a dog walker takes care of $n \geqslant 12$ dogs. The dog walker is not a strong person, and will walk dogs in groups of 3 or 7 at a time (every dog gets walked exactly once). Prove that the dog walker can always split the n dogs into groups of 3 dogs or 7 dogs.

Let $P(n)$ be "a group with n dogs can be split into groups of 3 dogs or 7 dogs." We will prove $P(n)$ for all natural numbers $n \geqslant 12$ by strong induction.

Base Cases $n = 12, 13, 14,$ **or** 15**:** 12 = 3 + 3 + 3 + 3, 13 = 3 + 7 + 3, 14 = 7 + 7, So $P(12)$, $P(13)$, and $P(14)$ hold.

Task 1 – Walk the Dawgs

Suppose that a dog walker takes care of $n \geqslant 12$ dogs. The dog walker is not a strong person, and will walk dogs in groups of 3 or 7 at a time (every dog gets walked exactly once). Prove that the dog walker can always split the n dogs into groups of 3 dogs or 7 dogs.

Let $P(n)$ be "a group with n dogs can be split into groups of 3 dogs or 7 dogs." We will prove $P(n)$ for all natural numbers $n \geqslant 12$ by strong induction.

Base Cases $n = 12, 13, 14,$ **or** 15**:** 12 = 3 + 3 + 3 + 3, 13 = 3 + 7 + 3, 14 = 7 + 7, So $P(12)$, $P(13)$, and $P(14)$ hold.

Inductive Hypothesis: Assume that $P(12), \ldots, P(k)$ hold for some arbitrary $k \geqslant$ 14.

Task 1 – Walk the Dawgs

Suppose that a dog walker takes care of $n \geqslant 12$ dogs. The dog walker is not a strong person, and will walk dogs in groups of 3 or 7 at a time (every dog gets walked exactly once). Prove that the dog walker can always split the n dogs into groups of 3 dogs or 7 dogs.

Let $P(n)$ be "a group with n dogs can be split into groups of 3 dogs or 7 dogs." We will prove $P(n)$ for all natural numbers $n \geqslant 12$ by strong induction.

Base Cases $n = 12, 13, 14,$ **or** 15**:** 12 = 3 + 3 + 3 + 3, 13 = 3 + 7 + 3, 14 = 7 + 7, So $P(12)$, $P(13)$, and $P(14)$ hold.

Inductive Hypothesis: Assume that $P(12), \ldots, P(k)$ hold for some arbitrary $k \geqslant 14$.

Inductive Step: Goal: Show $k + 1$ dogs can be split into groups of 3 dogs or 7 dogs.

Task 1 – Walk the Dawgs

Suppose that a dog walker takes care of $n \geqslant 12$ dogs. The dog walker is not a strong person, and will walk dogs in groups of 3 or 7 at a time (every dog gets walked exactly once). Prove that the dog walker can always split the n dogs into groups of 3 dogs or 7 dogs.

Let $P(n)$ be "a group with n dogs can be split into groups of 3 dogs or 7 dogs." We will prove $P(n)$ for all natural numbers $n \geqslant 12$ by strong induction.

Base Cases $n = 12, 13, 14,$ **or** 15**:** 12 = 3 + 3 + 3 + 3, 13 = 3 + 7 + 3, 14 = 7 + 7, So $P(12)$, $P(13)$, and $P(14)$ hold.

Inductive Hypothesis: Assume that $P(12), \ldots, P(k)$ hold for some arbitrary $k \geqslant 14$.

Inductive Step: Goal: Show $k + 1$ dogs can be split into groups of 3 dogs or 7 dogs.

We first form one group of 3 dogs out of the $k + 1$ dogs.

Task 1 – Walk the Dawgs

Suppose that a dog walker takes care of $n \geqslant 12$ dogs. The dog walker is not a strong person, and will walk dogs in groups of 3 or 7 at a time (every dog gets walked exactly once). Prove that the dog walker can always split the n dogs into groups of 3 dogs or 7 dogs.

Let $P(n)$ be "a group with n dogs can be split into groups of 3 dogs or 7 dogs." We will prove $P(n)$ for all natural numbers $n \geqslant 12$ by strong induction.

Base Cases $n = 12, 13, 14,$ **or** 15**:** 12 = 3 + 3 + 3 + 3, 13 = 3 + 7 + 3, 14 = 7 + 7, So $P(12)$, $P(13)$, and $P(14)$ hold.

Inductive Hypothesis: Assume that $P(12), \ldots, P(k)$ hold for some arbitrary $k \geqslant$ 14.

Inductive Step: Goal: Show $k + 1$ dogs can be split into groups of 3 dogs or 7 dogs.

We first form one group of 3 dogs out of the $k + 1$ dogs. Then we can divide the remaining $k - 2$ dogs into groups of 3 or 7 by the assumption $P(k - 2)$. (Note that $k \geqslant 14$ and so $k - 2 \geqslant 12$; thus, $P(k - 2)$ is among our assumptions $P(12), \ldots, P(k)$.)

Task 1 – Walk the Dawgs

Suppose that a dog walker takes care of $n \geqslant 12$ dogs. The dog walker is not a strong person, and will walk dogs in groups of 3 or 7 at a time (every dog gets walked exactly once). Prove that the dog walker can always split the n dogs into groups of 3 dogs or 7 dogs.

Let $P(n)$ be "a group with n dogs can be split into groups of 3 dogs or 7 dogs." We will prove $P(n)$ for all natural numbers $n \geqslant 12$ by strong induction.

Base Cases $n = 12, 13, 14,$ **or** 15**:** 12 = 3 + 3 + 3 + 3, 13 = 3 + 7 + 3, 14 = 7 + 7, So $P(12)$, $P(13)$, and $P(14)$ hold.

Inductive Hypothesis: Assume that $P(12), \ldots, P(k)$ hold for some arbitrary $k \geqslant 14$.

Inductive Step: Goal: Show $k + 1$ dogs can be split into groups of 3 dogs or 7 dogs.

We first form one group of 3 dogs out of the $k + 1$ dogs. Then we can divide the remaining $k - 2$ dogs into groups of 3 or 7 by the assumption $P(k - 2)$. (Note that $k \geqslant 14$ and so $k - 2 \geqslant 12$; thus, $P(k - 2)$ is among our assumptions $P(12), \ldots, P(k)$.)

Conclusion: $P(n)$ holds for all integers $n \geqslant 12$ by by principle of strong induction.

Structural Induction

Idea of Structural Induction

Every element is built up recursively…

So to show $P(s)$ for all $s \in S$…

Show $P(b)$ for all base case elements b.

Show for an arbitrary element not in the base case, if $P()$ holds for every named element in the recursive rule, then $P()$ holds for the new element (each recursive rule will be a case of this proof).

Structural Induction Template

Let $P(x)$ be "<predicate>". We show $P(x)$ holds for all $x \in S$ by structural induction.

Base Case: Show $P(x)$
[Do that for every base cases x in S.]

Inductive Hypothesis: Suppose $P(x)$ for an arbitrary x
[Do that for every x listed as in S in the recursive rules.]

Inductive Step: Show $P()$ holds for y.
[You will need a separate case/step for every recursive rule.]

Therefore $P(x)$ holds for all $x \in S$ by the principle of induction.

Problem 2 - Structural Induction on Strings

Definition of string:
Basis Step: "" is a string.
Recursive Step: If X is a string and c is a character then append(c, X) is a string.

Definition of **len**():
len("") = 0
len(append(c, X)) = 1 + len(X)

Definition of **double**():
double("") = ""
double(append(c, X)) = append(c, append(c, double(X)))

Prove that for any string X, len(double(X)) = 2len(X).

Problem 2 - Structural Induction on Strings

Prove that for any string X,
len(double(X)) = 2len(X)

For x ∈ S, let P(x) be "".
We show P(x) holds for all x ∈ S by structural induction on x.

Base Case: Show P(x) (for all x in the basis rules)

Inductive Hypothesis: Suppose P(x) (for all x in the recursive rules),
i.e. (IH in terms of P(x))

Inductive Step: Goal: Show that P(?) holds. (IS goal in terms of P(?))

Conclusion: Therefore P(x) holds for all x ∈ S by structural induction.

Problem 2 - Structural Induction on Strings

Prove that for any string X,
len(double(X)) = 2len(X)

For a string X, let P(X) be "len(double(X)) = 2len(X)".
We show P(x) holds for all x ∈ S by structural induction on x.

Base Case: Show P(x) (for all x in the basis rules)

Inductive Hypothesis: Suppose P(x) (for all x in the recursive rules),
i.e. (IH in terms of P(x))

Inductive Step: Goal: Show that P(?) holds. (IS goal in terms of P(?))

Conclusion: Therefore P(x) holds for all x ∈ S by structural induction.

Problem 2 - Structural Induction on Strings

Prove that for any string X,
len(double(X)) = 2len(X)

For a string X, let P(X) be “len(double(X)) = 2len(X)”.
We prove P(X) for all strings X by structural induction on X

<u>Base Case:</u> Show P(x) (for all x in the basis rules)

<u>Inductive Hypothesis:</u> Suppose P(x) (for all x in the recursive rules), i.e. (IH in terms of P(x))

<u>Inductive Step:</u> Goal: Show that P(?) holds. (IS goal in terms of P(?))

<u>Conclusion:</u> Therefore P(x) holds for all x ∈ S by structural induction.

Problem 2 - Structural Induction on Strings

Prove that for any string X,
len(double(X)) = 2len(X)

For a string X, let P(X) be “len(double(X)) = 2len(X)”.
We prove P(X) for all strings X by structural induction on X

Base Case: P(""): By definition, len(double("")) = len("") = 0 = 2 · 0 = 2len(""), so P("") holds

Inductive Hypothesis: Suppose P(x) (for all x in the recursive rules),
i.e. (IH in terms of P(x))

Inductive Step: Goal: Show that P(?) holds. (IS goal in terms of P(?))

Conclusion: Therefore P(x) holds for all x ∈ S by structural induction.

Problem 2 - Structural Induction on Strings

Prove that for any string X,
len(double(X)) = 2len(X)

For a string X, let P(X) be "len(double(X)) = 2len(X)".
We prove P(X) for all strings X by structural induction on X

Base Case: P(""): By definition, len(double("")) = len("") = 0 = 2 · 0 = 2len(""), so P("") holds

Inductive Hypothesis: Suppose P(X) holds for some arbitrary string X,
i.e. (IH in terms of P(x))

Inductive Step: Goal: Show that P(?) holds. (IS goal in terms of P(?))

Conclusion: Therefore P(x) holds for all x ∈ S by structural induction.

Problem 2 - Structural Induction on Strings

Prove that for any string X,
len(double(X)) = 2len(X)

For a string X, let P(X) be “len(double(X)) = 2len(X)”.
We prove P(X) for all strings X by structural induction on X

<u>Base Case:</u> P(""): By definition, len(double("")) = len("") = 0 = 2 · 0 = 2len(""), so P("") holds

<u>Inductive Hypothesis:</u> Suppose P(X) holds for some arbitrary string X,
i.e. len(double(X)) = 2len(X)

<u>Inductive Step:</u> Goal: Show that P(?) holds. (IS goal in terms of P(?))

<u>Conclusion:</u> Therefore P(x) holds for all x ∈ S by structural induction.

Problem 2 - Structural Induction on Strings

Prove that for any string X,
len(double(X)) = 2len(X)

For a string X, let P(X) be "len(double(X)) = 2len(X)".
We prove P(X) for all strings X by structural induction on X

Base Case: P(""): By definition, len(double("")) = len("") = 0 = 2 · 0 = 2len(""), so P("") holds

Inductive Hypothesis: Suppose P(X) holds for some arbitrary string X,
i.e. len(double(X)) = 2len(X)

Inductive Step: Goal: Show P(append(c, X)) for any c: len(double(append(c, X))) = 2(len(append(c, X)))

Conclusion: Therefore P(x) holds for all x ∈ S by structural induction

Problem 2 - Structural Induction on Strings

Prove that for any string X,
len(double(X)) = 2len(X)

For a string X, let P(X) be “len(double(X)) = 2len(X)”.
We prove P(X) for all strings X by structural induction on X

<u>Base Case:</u> P(""): By definition, len(double("")) = len("") = 0 = 2 · 0 = 2len(""), so P("") holds

<u>Inductive Hypothesis:</u> Suppose P(X) holds for some arbitrary string X,
i.e. len(double(X)) = 2len(X)

<u>Inductive Step:</u> Goal: Show P(append(c, X)) for any c: len(double(append(c, X))) = 2(len(append(c, X)))

Conclusion: Therefore P(X) holds for all strings X by structural induction

Problem 2 - Structural Induction on Strings

Prove that for any string X,
len(double(X)) = 2len(X)

For a string X, let P(X) be “len(double(X)) = 2len(X)”.
We prove P(X) for all strings X by structural induction on X

Base Case: P(""): By definition, len(double("")) = len("") = 0 = 2 · 0 = 2len(""), so P("") holds

Inductive Hypothesis: Suppose P(X) holds for some arbitrary string X,
i.e. len(double(X)) = 2len(X)

Inductive Step: Goal: Show P(append(c, X)) for any c: len(double(append(c, X))) = 2(len(append(c, X)))

len(double(append(c, X))) =

Conclusion: Therefore P(X) holds for all strings X by structural induction

Problem 2 - Structural Induction on Strings

Prove that for any string X,
len(double(X)) = 2len(X)

For a string X, let P(X) be “len(double(X)) = 2len(X)”.
We prove P(X) for all strings X by structural induction on X

<u>Base Case:</u> P(""): By definition, len(double("")) = len("") = 0 = 2 · 0 = 2len(""), so P("") holds

<u>Inductive Hypothesis:</u> Suppose P(X) holds for some arbitrary string X,
i.e. len(double(X)) = 2len(X)

<u>Inductive Step:</u> Goal: Show P(append(c, X)) for any c: len(double(append(c, X))) = 2(len(append(c, X)))

len(double(append(c, X))) = len(append(c, append(c, double(X)))) definition of double

Conclusion: Therefore P(X) holds for all strings X by structural induction

Problem 2 - Structural Induction on Strings

Prove that for any string X,
len(double(X)) = 2len(X)

For a string X, let P(X) be "len(double(X)) = 2len(X)".
We prove P(X) for all strings X by structural induction on X

Base Case: P(""): By definition, len(double("")) = len("") = 0 = 2 · 0 = 2len(""), so P("") holds

Inductive Hypothesis: Suppose P(X) holds for some arbitrary string X,
i.e. len(double(X)) = 2len(X)

Inductive Step: Goal: Show P(append(c, X)) for any c: len(double(append(c, X))) = 2(len(append(c, X)))

len(double(append(c, X)))	= len(append(c, append(c, double(X))))	definition of double
	= 1 + len(append(c, double(X)))	definition of len

Conclusion: Therefore P(X) holds for all strings X by structural induction

Problem 2 - Structural Induction on Strings

Prove that for any string X,
len(double(X)) = 2len(X)

For a string X, let P(X) be “len(double(X)) = 2len(X)”.
We prove P(X) for all strings X by structural induction on X

<u>Base Case:</u> P(""): By definition, len(double("")) = len("") = 0 = 2 · 0 = 2len(""), so P("") holds

<u>Inductive Hypothesis:</u> Suppose P(X) holds for some arbitrary string X,
i.e. len(double(X)) = 2len(X)

<u>Inductive Step:</u> Goal: Show P(append(c, X)) for any c: len(double(append(c, X))) = 2(len(append(c, X)))

len(double(append(c, X))) = len(append(c, append(c, double(X))))   definition of double
= 1 + len(append(c, double(X)))   definition of len
= 1 + 1 + len(double(X))   definition of len

Conclusion: Therefore P(X) holds for all strings X by structural induction

Problem 2 - Structural Induction on Strings

Prove that for any string X,
len(double(X)) = 2len(X)

For a string X, let P(X) be “len(double(X)) = 2len(X)”.
We prove P(X) for all strings X by structural induction on X

<u>Base Case:</u> P(""): By definition, len(double("")) = len("") = 0 = 2 · 0 = 2len(""), so P("") holds

<u>Inductive Hypothesis:</u> Suppose P(X) holds for some arbitrary string X,
i.e. len(double(X)) = 2len(X)

<u>Inductive Step:</u> Goal: Show P(append(c, X)) for any c: len(double(append(c, X))) = 2(len(append(c, X)))

len(double(append(c, X)))	= len(append(c, append(c, double(X))))	definition of double
	= 1 + len(append(c, double(X)))	definition of len
	= 1 + 1 + len(double(X))	definition of len
	= 2 + 2len(X)	by I.H.

Conclusion: Therefore P(X) holds for all strings X by structural induction

Problem 2 - Structural Induction on Strings

Prove that for any string X,
len(double(X)) = 2len(X)

For a string X, let P(X) be "len(double(X)) = 2len(X)".
We prove P(X) for all strings X by structural induction on X

<u>Base Case:</u> P(""): By definition, len(double("")) = len("") = 0 = 2 · 0 = 2len(""), so P("") holds

<u>Inductive Hypothesis:</u> Suppose P(X) holds for some arbitrary string X,
i.e. len(double(X)) = 2len(X)

<u>Inductive Step:</u> Goal: Show P(append(c, X)) for any c: len(double(append(c, X))) = 2(len(append(c, X)))

len(double(append(c, X)))	= len(append(c, append(c, double(X))))	definition of double
	= 1 + len(append(c, double(X)))	definition of len
	= 1 + 1 + len(double(X))	definition of len
	= 2 + 2len(X)	by I.H.
	= 2(1 + len(X))	

<u>Conclusion:</u> Therefore P(X) holds for all strings X by structural induction.

Problem 2 - Structural Induction on Strings

Prove that for any string X,
len(double(X)) = 2len(X)

For a string X, let P(X) be “len(double(X)) = 2len(X)”.
We prove P(X) for all strings X by structural induction on X

<u>Base Case:</u> P(""): By definition, len(double("")) = len("") = 0 = 2 · 0 = 2len(""), so P("") holds

<u>Inductive Hypothesis:</u> Suppose P(X) holds for some arbitrary string X,
i.e. len(double(X)) = 2len(X)

<u>Inductive Step:</u> Goal: Show P(append(c, X)) for any c: len(double(append(c, X))) = 2(len(append(c, X)))

len(double(append(c, X)))	= len(append(c, append(c, double(X))))	definition of double
	= 1 + len(append(c, double(X)))	definition of len
	= 1 + 1 + len(double(X))	definition of len
	= 2 + 2len(X)	by I.H.
	= 2(1 + len(X))	
	= 2(len(append(c, X)))	definition of len

<u>Conclusion:</u> Therefore P(X) holds for all strings X by structural induction.

Problem 2 - Structural Induction on Strings

Prove that for any string X,
len(double(X)) = 2len(X)

For a string X, let P(X) be “len(double(X)) = 2len(X)”.
We prove P(X) for all strings X by structural induction on X

Base Case: P(""): By definition, len(double("")) = len("") = 0 = 2 · 0 = 2len(""), so P("") holds

Inductive Hypothesis: Suppose P(X) holds for some arbitrary string X,
i.e. len(double(X)) = 2len(X)

Inductive Step: Goal: Show P(append(c, X)) for any c: len(double(append(c, X))) = 2(len(append(c, X)))

len(double(append(c, X)))	= len(append(c, append(c, double(X))))	definition of double
	= 1 + len(append(c, double(X)))	definition of len
	= 1 + 1 + len(double(X))	definition of len
	= 2 + 2len(X)	by I.H.
	= 2(1 + len(X))	
	= 2(len(append(c, X)))	definition of len

So, P(append(c, X)) holds!

Conclusion: Therefore P(X) holds for all strings X by structural induction.

Problem 3 – Structural Induction on Trees

Definition of Tree:
Basis Step: • is a Tree.
Recursive Step: If L is a Tree and R is a Tree then Tree(•, L, R) is a Tree

Definition of leaves():
leaves(•) = 1
leaves(Tree(•, L, R)) = leaves(L) + leaves(R)

Definition of size():
size(•) = 1
size(Tree(•, L, R)) =1 + size(L) + size(R)

Prove that leaves(T) ≥ size(T)/2 + 1/2 for all Trees T

Work on this problem with the people around you.

Problem 3 - Structural Induction on Trees

Prove that leaves(T) ≥ size(T)/2 + 1/2 for all Trees T

For x ∈ S, let P(x) be “”.
We show P(x) holds for all x ∈ S by structural induction on x.

<u>Base Case:</u> Show P(x) (for all x in the basis rules)

<u>Inductive Hypothesis:</u> Suppose P(x) (for all x in the recursive rules), i.e. (IH in terms of P(x))

<u>Inductive Step:</u> Goal: Show that P(?) holds. (IS goal in terms of P(?))

<u>Conclusion:</u> Therefore P(x) holds for all x ∈ S by the principle of induction.

Problem 3 - Structural Induction on Trees

Prove that leaves(T) ≥ size(T)/2 + 1/2 for all Trees T

For a tree T, let P(T) be "leaves(T) ≥ size(T)/2 + 1/2".
We show P(x) holds for all x ∈ S by structural induction on x.

Base Case: Show P(x) (for all x in the basis rules)

Inductive Hypothesis: Suppose P(x) (for all x in the recursive rules),
i.e. (IH in terms of P(x))

Inductive Step: Goal: Show that P(?) holds. (IS goal in terms of P(?))

Conclusion: Therefore P(x) holds for all x ∈ S by the principle of induction.

Problem 3 - Structural Induction on Trees

Prove that leaves(T) ≥ size(T)/2 + 1/2 for all Trees T

For a tree T, let P(T) be "leaves(T) ≥ size(T)/2 + 1/2".
We show P(T) holds for all trees T by structural induction on T.

Base Case: Show P(x) (for all x in the basis rules)

Inductive Hypothesis: Suppose P(x) (for all x in the recursive rules),
i.e. (IH in terms of P(x))

Inductive Step: Goal: Show that P(?) holds. (IS goal in terms of P(?))

Conclusion: Therefore P(x) holds for all x ∈ S by the principle of induction.

Problem 3 - Structural Induction on Trees

Prove that leaves(T) ≥ size(T)/2 + 1/2 for all Trees T

For a tree T, let P(T) be "leaves(T) ≥ size(T)/2 + 1/2".
We show P(T) holds for all trees T by structural induction on T.

Base Case: P(•): By definition of leaves(•), leaves(•) = 1 and size(•) = 1.
So, leaves(•) = 1 ≥ 1/2 + 1/2 = size(•)/2 + 1/2, so P(•) holds.

Inductive Hypothesis: Suppose P(x) (for all x in the recursive rules),
i.e. (IH in terms of P(x))

Inductive Step: Goal: Show that P(?) holds. (IS goal in terms of P(?))

Conclusion: Therefore P(x) holds for all x ∈ S by the principle of induction.

Problem 3 - Structural Induction on Trees

Prove that leaves(T) ≥ size(T)/2 + 1/2 for all Trees T

For a tree T, let P(T) be "leaves(T) ≥ size(T)/2 + 1/2".
We show P(T) holds for all trees T by structural induction on T.

Base Case: P(•): By definition of leaves(•), leaves(•) = 1 and size(•) = 1.
So, leaves(•) = 1 ≥ 1/2 + 1/2 = size(•)/2 + 1/2, so P(•) holds.

Inductive Hypothesis: Suppose P(L) and P(R) hold for some arbitrary trees L and R,
i.e. (IH in terms of P(x))

Inductive Step: Goal: Show that P(?) holds. (IS goal in terms of P(?))

Conclusion: Therefore P(x) holds for all x ∈ S by the principle of induction.

Problem 3 - Structural Induction on Trees

Prove that leaves(T) ≥ size(T)/2 + 1/2 for all Trees T

For a tree T, let P(T) be "leaves(T) ≥ size(T)/2 + 1/2".
We show P(T) holds for all trees T by structural induction on T.

<u>Base Case:</u> P(•): By definition of leaves(•), leaves(•) = 1 and size(•) = 1.
So, leaves(•) = 1 ≥ 1/2 + 1/2 = size(•)/2 + 1/2, so P(•) holds.

<u>Inductive Hypothesis:</u> Suppose P(L) and P(R) hold for some arbitrary trees L and R,
i.e., leaves(L) ≥ size(L)/2 + 1/2, leaves(R) ≥ size(R)/2 + 1/2

<u>Inductive Step:</u> Goal: Show that P(?) holds. (IS goal in terms of P(?))

<u>Conclusion:</u> Therefore P(x) holds for all x ∈ S by the principle of induction.

Problem 3 - Structural Induction on Trees

Prove that leaves(T) ≥ size(T)/2 + 1/2 for all Trees T

For a tree T, let P(T) be "leaves(T) ≥ size(T)/2 + 1/2".
We show P(T) holds for all trees T by structural induction on T.

Base Case: P(•): By definition of leaves(•), leaves(•) = 1 and size(•) = 1.
So, leaves(•) = 1 ≥ 1/2 + 1/2 = size(•)/2 + 1/2, so P(•) holds.

Inductive Hypothesis: Suppose P(L) and P(R) hold for some arbitrary trees L and R,
i.e., leaves(L) ≥ size(L)/2 + 1/2, leaves(R) ≥ size(R)/2 + 1/2

Inductive Step: Goal: Show P(**Tree(•, L, R)**): leaves(**Tree(•, L, R)**) ≥ size(**Tree(•, L, R)**)/2 + 1/2

Conclusion: Therefore P(x) holds for all x ∈ S by the principle of induction.

Problem 3 - Structural Induction on Trees

Prove that leaves(T) ≥ size(T)/2 + 1/2 for all Trees T

For a tree T, let P(T) be “leaves(T) ≥ size(T)/2 + 1/2”.
We show P(T) holds for all trees T by structural induction on T.

Base Case: P(•): By definition of leaves(•), leaves(•) = 1 and size(•) = 1.
So, leaves(•) = 1 ≥ 1/2 + 1/2 = size(•)/2 + 1/2, so P(•) holds.

Inductive Hypothesis: Suppose P(L) and P(R) hold for some arbitrary trees L and R,
i.e., leaves(L) ≥ size(L)/2 + 1/2, leaves(R) ≥ size(R)/2 + 1/2

Inductive Step: Goal: Show P(**Tree(•, L, R)**): leaves(**Tree(•, L, R)**) ≥ size(**Tree(•, L, R)**)/2 + 1/2

Conclusion: Therefore P(T) holds for all trees T by the principle of induction.

Problem 3 - Structural Induction on Trees

Prove that leaves(T) ≥ size(T)/2 + 1/2 for all Trees T

For a tree T, let P(T) be “leaves(T) ≥ size(T)/2 + 1/2”.
We show P(T) holds for all trees T by structural induction on T.

Base Case: P(•): By definition of leaves(•), leaves(•) = 1 and size(•) = 1.
So, leaves(•) = 1 ≥ 1/2 + 1/2 = size(•)/2 + 1/2, so P(•) holds.

Inductive Hypothesis: Suppose P(L) and P(R) hold for some arbitrary trees L and R,
i.e., leaves(L) ≥ size(L)/2 + 1/2, leaves(R) ≥ size(R)/2 + 1/2

Inductive Step: Goal: Show P(**Tree(•, L, R)**): leaves(**Tree(•, L, R)**) ≥ size(**Tree(•, L, R)**)/2 + 1/2

Again, as long as you can get this far, you will get the majority of points on the problem! Go for this skeleton first, and then think about what you need to do to complete the proof.

Conclusion: Therefore P(T) holds for all trees T by the principle of induction.

Problem 3 - Structural Induction on Trees

Prove that leaves(T) ≥ size(T)/2 + 1/2 for all Trees T

For a tree T, let P(T) be "leaves(T) ≥ size(T)/2 + 1/2".
We show P(T) holds for all trees T by structural induction on T.

Base Case: P(•): By definition of leaves(•), leaves(•) = 1 and size(•) = 1.
So, leaves(•) = 1 ≥ 1/2 + 1/2 = size(•)/2 + 1/2, so P(•) holds.

Inductive Hypothesis: Suppose P(L) and P(R) hold for some arbitrary trees L and R,
i.e., leaves(L) ≥ size(L)/2 + 1/2, leaves(R) ≥ size(R)/2 + 1/2

Inductive Step: Goal: Show P(**Tree(•, L, R)**): leaves(**Tree(•, L, R)**) ≥ size(**Tree(•, L, R)**)/2 + 1/2

leaves(**Tree(•, L, R)**) =

Conclusion: Therefore P(T) holds for all trees T by the principle of induction.

Problem 3 - Structural Induction on Trees

Prove that leaves(T) ≥ size(T)/2 + 1/2 for all Trees T

For a tree T, let P(T) be "leaves(T) ≥ size(T)/2 + 1/2".
We show P(T) holds for all trees T by structural induction on T.

<u>Base Case:</u> P(•): By definition of leaves(•), leaves(•) = 1 and size(•) = 1.
So, leaves(•) = 1 ≥ 1/2 + 1/2 = size(•)/2 + 1/2, so P(•) holds.

<u>Inductive Hypothesis:</u> Suppose P(L) and P(R) hold for some arbitrary trees L and R,
i.e., leaves(L) ≥ size(L)/2 + 1/2, leaves(R) ≥ size(R)/2 + 1/2

<u>Inductive Step:</u> Goal: Show P(**Tree(•, L, R)**): leaves(**Tree(•, L, R)**) ≥ size(**Tree(•, L, R)**)/2 + 1/2

leaves(**Tree(•, L, R)**) = leaves(**L**) + leaves(**R**)	definition of leaves

<u>Conclusion:</u> Therefore P(T) holds for all trees T by the principle of induction.

Problem 3 - Structural Induction on Trees

Prove that leaves(T) ≥ size(T)/2 + 1/2 for all Trees T

For a tree T, let P(T) be “leaves(T) ≥ size(T)/2 + 1/2”.
We show P(T) holds for all trees T by structural induction on T.

<u>Base Case:</u> P(•): By definition of leaves(•), leaves(•) = 1 and size(•) = 1.
So, leaves(•) = 1 ≥ 1/2 + 1/2 = size(•)/2 + 1/2, so P(•) holds.

<u>Inductive Hypothesis:</u> Suppose P(L) and P(R) hold for some arbitrary trees L and R,
i.e., leaves(L) ≥ size(L)/2 + 1/2, leaves(R) ≥ size(R)/2 + 1/2

<u>Inductive Step:</u> Goal: Show P(**Tree(•, L, R)**): leaves(**Tree(•, L, R)**) ≥ size(**Tree(•, L, R)**)/2 + 1/2

leaves(**Tree(•, L, R)**)	= leaves(**L**) + leaves(**R**)	definition of leaves
	≥ (size(**L**)/2 + 1/2) + (size(**R**)/2 + 1/2)	by Inductive Hypothesis

<u>Conclusion:</u> Therefore P(T) holds for all trees T by the principle of induction.

Problem 3 - Structural Induction on Trees

Prove that leaves(T) ≥ size(T)/2 + 1/2 for all Trees T

For a tree T, let P(T) be "leaves(T) ≥ size(T)/2 + 1/2".
We show P(T) holds for all trees T by structural induction on T.

Base Case: P(•): By definition of leaves(•), leaves(•) = 1 and size(•) = 1.
So, leaves(•) = 1 ≥ 1/2 + 1/2 = size(•)/2 + 1/2, so P(•) holds.

Inductive Hypothesis: Suppose P(L) and P(R) hold for some arbitrary trees L and R,
i.e., leaves(L) ≥ size(L)/2 + 1/2, leaves(R) ≥ size(R)/2 + 1/2

Inductive Step: Goal: Show P(**Tree(•, L, R)**): leaves(**Tree(•, L, R)**) ≥ size(**Tree(•, L, R)**)/2 + 1/2

leaves(**Tree(•, L, R)**)	= leaves(**L**) + leaves(**R**)	definition of leaves
	≥ (size(**L**)/2 + 1/2) + (size(**R**)/2 + 1/2)	by Inductive Hypothesis
	= (1/2 + size(**L**)/2 + size(**R**)/2) + 1/2	

Conclusion: Therefore P(T) holds for all trees T by the principle of induction.

Problem 3 - Structural Induction on Trees

Prove that leaves(T) ≥ size(T)/2 + 1/2 for all Trees T

For a tree T, let P(T) be "leaves(T) ≥ size(T)/2 + 1/2".
We show P(T) holds for all trees T by structural induction on T.

<u>Base Case:</u> P(•): By definition of leaves(•), leaves(•) = 1 and size(•) = 1.
So, leaves(•) = 1 ≥ 1/2 + 1/2 = size(•)/2 + 1/2, so P(•) holds.

<u>Inductive Hypothesis:</u> Suppose P(L) and P(R) hold for some arbitrary trees L and R,
i.e., leaves(L) ≥ size(L)/2 + 1/2, leaves(R) ≥ size(R)/2 + 1/2

<u>Inductive Step:</u> Goal: Show P(**Tree(•, L, R)**): leaves(**Tree(•, L, R)**) ≥ size(**Tree(•, L, R)**)/2 + 1/2

leaves(**Tree(•, L, R)**)	= leaves(**L**) + leaves(**R**)	definition of leaves
	≥ (size(**L**)/2 + 1/2) + (size(**R**)/2 + 1/2)	by Inductive Hypothesis
	= (1/2 + size(**L**)/2 + size(**R**)/2) + 1/2	
	= (1 + size(**L**) + size(**R**)) / 2 + 1/2	

<u>Conclusion:</u> Therefore P(T) holds for all trees T by the principle of induction.

Problem 3 - Structural Induction on Trees

Prove that leaves(T) ≥ size(T)/2 + 1/2 for all Trees T

For a tree T, let P(T) be "leaves(T) ≥ size(T)/2 + 1/2".
We show P(T) holds for all trees T by structural induction on T.

Base Case: P(•): By definition of leaves(•), leaves(•) = 1 and size(•) = 1.
So, leaves(•) = 1 ≥ 1/2 + 1/2 = size(•)/2 + 1/2, so P(•) holds.

Inductive Hypothesis: Suppose P(L) and P(R) hold for some arbitrary trees L and R,
i.e., leaves(L) ≥ size(L)/2 + 1/2, leaves(R) ≥ size(R)/2 + 1/2

Inductive Step: Goal: Show P(**Tree(•, L, R)**): leaves(**Tree(•, L, R)**) ≥ size(**Tree(•, L, R)**)/2 + 1/2

leaves(**Tree(•, L, R)**) = leaves(**L**) + leaves(**R**) — definition of leaves
≥ (size(**L**)/2 + 1/2) + (size(**R**)/2 + 1/2) — by Inductive Hypothesis
= (1/2 + size(**L**)/2 + size(**R**)/2) + 1/2
= (1 + size(**L**) + size(**R**)) / 2 + 1/2
= size(**Tree(•, L, R)**)/2 + 1/2 — definition of size

Conclusion: Therefore P(T) holds for all trees T by the principle of induction.

Problem 3 - Structural Induction on Trees

Prove that leaves(T) ≥ size(T)/2 + 1/2 for all Trees T

For a tree T, let P(T) be "leaves(T) ≥ size(T)/2 + 1/2".
We show P(T) holds for all trees T by structural induction on T.

<u>Base Case:</u> P(•): By definition of leaves(•), leaves(•) = 1 and size(•) = 1.
So, leaves(•) = 1 ≥ 1/2 + 1/2 = size(•)/2 + 1/2, so P(•) holds.

<u>Inductive Hypothesis:</u> Suppose P(L) and P(R) hold for some arbitrary trees L and R,
i.e., leaves(L) ≥ size(L)/2 + 1/2, leaves(R) ≥ size(R)/2 + 1/2

<u>Inductive Step:</u> Goal: Show P(Tree(•, L, R)): leaves(Tree(•, L, R)) ≥ size(Tree(•, L, R))/2 + 1/2

leaves(**Tree(•, L, R)**)	= leaves(**L**) + leaves(**R**)	definition of leaves
	≥ (size(**L**)/2 + 1/2) + (size(**R**)/2 + 1/2)	by Inductive Hypothesis
	= (1/2 + size(**L**)/2 + size(**R**)/2) + 1/2	
	= (1 + size(**L**) + size(**R**)) / 2 + 1/2	
	= size(**Tree(•, L, R)**)/2 + 1/2	definition of size

So, P(**Tree(•, L, R)**) holds!

<u>Conclusion:</u> Therefore P(T) holds for all trees T by the principle of induction.

That’s All, Folks!

Thanks for coming to section this week!
Any questions?