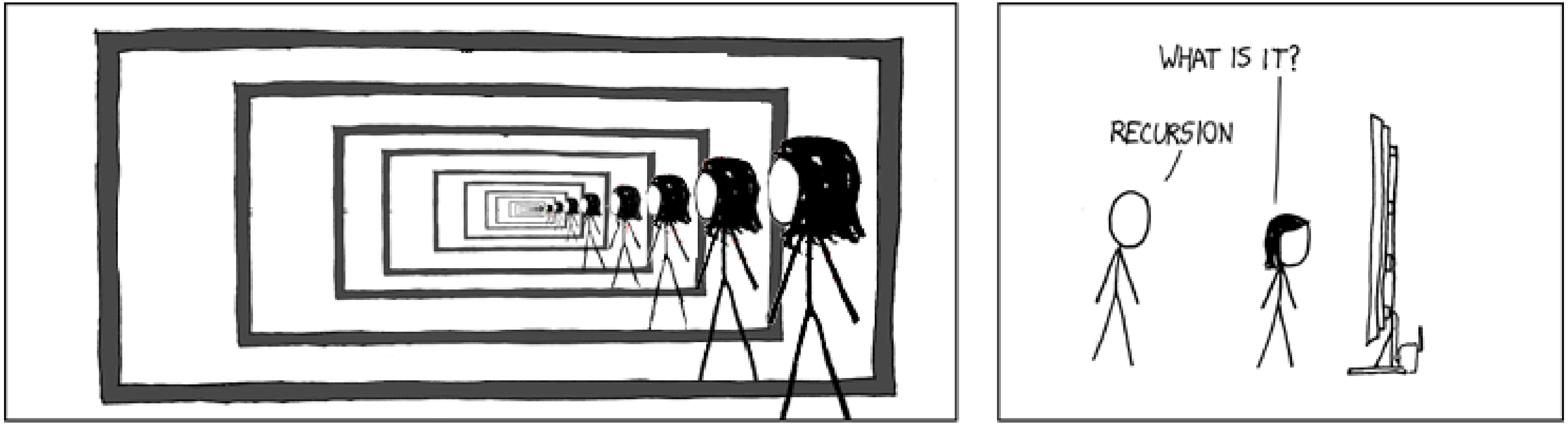


Structural Induction Cont.

CSE 311: Foundations of Computing I
Lecture 18

Announcements

- HW5 solutions are at the front.
- HW6 will be released today, due next Wednesday at 11:59pm on Gradescope.
- The Midterm has been graded, and scores will be published on Gradescope this afternoon

Midterm Grades

Overall we were very impressed with your work.

Stats:

- Median: 72 / 80 (90%)
- Mean: 66.58 / 80 (83.23%)

Course Grades

- The Midterm is only 15% of your overall course grade.
- Course grades are based on historical benchmarks for CSE 311.
  - If scores are lower than in the past, we will round grades up
  - If scores are higher than in the past, then we won't adjust down!
- The typical median GPA for this course is around a 3.5
- If you are concerned about your grade in the course, please feel free to reach out to anjalia@uw.edu to schedule an appointment.

Midterm Corrections

Midterm Corrections

- The purpose of an exam is to demonstrate learning
- We are offering the opportunity for you to correct your Midterm to earn a better score

Midterm Corrections

Detailed instructions will be posted on the course website under "Assignments" after class.

To submit corrections:

1. For each problem part that you wish to correct, rewrite your complete solution.
2. For each problem part that you wish to correct, write 2-4 sentences reflecting on **what** was incorrect in the original attempt, **why** the original error was made, and **how** you have fixed the error in your solutions.
3. Submit your corrections to Gradescope under "Midterm Corrections" by Wednesday, August 9th at 11:59pm.

Midterm Corrections

You **do not need** to submit corrections to every problem.

Your effective midterm score will be calculated as follows:

$$\text{midterm\_grade} = 0.75 \cdot \text{original\_grade} + 0.25 \cdot \max(\text{corrected\_grade}, \text{original\_grade})$$

Student A: 60% on the midterm, 95% after corrections. Effective grade is 68.75%.

Student B: 80% on the midterm, 70% after corrections. Effective grade is 80%.

Midterm Corrections

- You may not use Late Days on the Midterm corrections.
  - We intend to pass out Midterm solutions in class next Friday, August 11th
- Submissions without reflections will not be graded.
- The collaboration policy is the same as homework.
  - That means you may discuss with others and ask questions in OH, but you must write up solutions on your own
- Corrections will not be offered for the Final exam.

Recursively Defined Sets

Recursive Definition for Strings

Σ is the alphabet
Σ^* is the set of all strings
ε is the empty string

Basis Step: $\varepsilon \in \Sigma^*$

Recursive Step: If $w \in \Sigma^*$ and $a \in \Sigma$, then $wa \in \Sigma^*$

Recursive Definition for Lists of Integers

Basis Step: $[\,] \in \mathrm{List}$

Recursive Step: If $\mathrm{L} \in \mathrm{List}$ and $a \in \mathbb{Z}$ then $a :: \mathrm{L} \in \mathrm{List}$

Recursive Definition for Binary Trees of Integers

Basis Step: $\textbf{null} \in \text{Tree}$

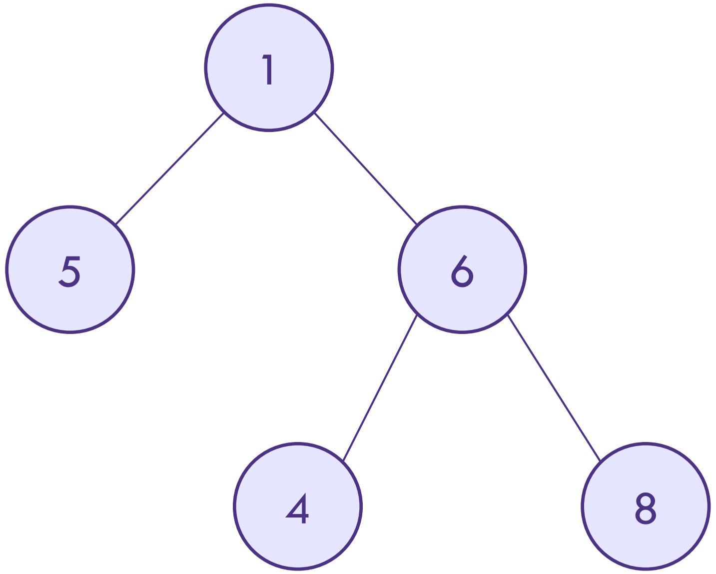
Recursive Step: If $L, R \in \text{Tree}$, and $a \in \mathbb{Z}$ then $(L, a, R) \in \text{Tree}$

Exercise

Basis: $\text{null} \in \text{Tree}$

Recursive: If $L, R \in \text{Tree}$, and $a \in \mathbb{Z}$ then $(L, a, R) \in \text{Tree}$

Write out the following tree using our notation for trees.



Claim 1

Functions on Binary Trees

Basis: $\text{null} \in \text{Tree}$

Recursive: If $L, R \in \text{Tree}$, and $a \in \mathbb{Z}$ then $(L, a, R) \in \text{Tree}$

To prove interesting facts about trees, we need functions on trees.

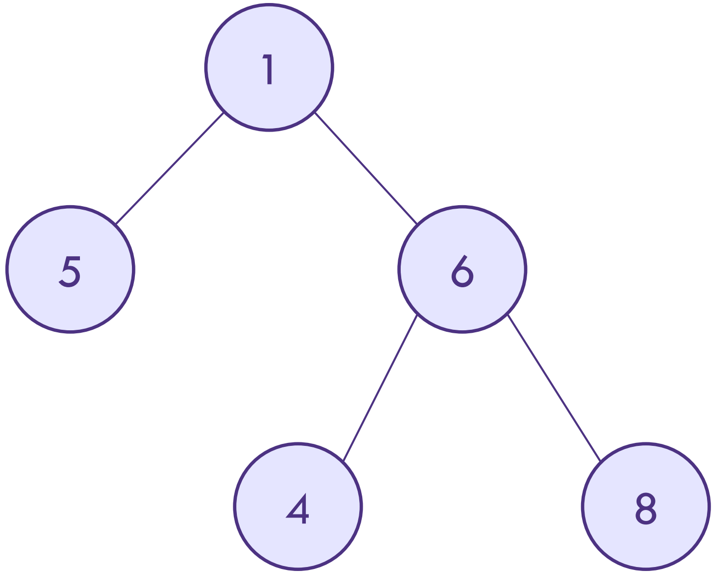
Size:

Height:

Functions on Binary Trees

$\text{size}(\text{null}) = 0$
$\text{size}\big((L, a, R)\big) = 1 + \text{size}(L) + \text{size}(R)$

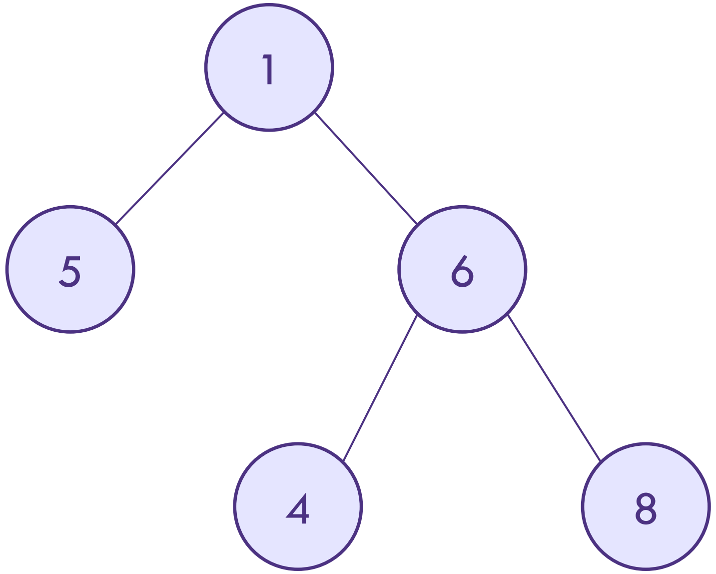
What's the size the tree $((\text{null}, 5, \text{null}), 1, ((\text{null}, 4, \text{null}), 6, (\text{null}, 8, \text{null})))$?



Functions on Binary Trees

$\text{height}(\text{null}) = -1$
$\text{height}\big((L, a, R)\big) = 1 + \max\big(\text{height}(L), \text{height}(R)\big)$

What's the height of the tree $((\text{null}, 5, \text{null}), 1, ((\text{null}, 4, \text{null}), 6, (\text{null}, 8, \text{null})))$?



Claim 1

$\text{height}(\text{null}) = -1$
$\text{height}\big((L, a, R)\big) = 1 + \max\big(\text{height}(L), \text{height}(R)\big)$

$\text{size}(\text{null}) = 0$
$\text{size}\big((L, a, R)\big) = 1 + \text{size}(L) + \text{size}(R)$

Claim 1: For every binary tree, $\text{size}(T) \leq 2^{\text{height}(T)+1} - 1$.

$\text{size}(\text{null}) = 0$
$\text{size}((L, a, R)) = 1 + \text{size}(L) + \text{size}(R)$

$\text{height}(\text{null}) = -1$
$\text{height}((L, a, R)) = 1 + \max(\text{height}(L), \text{height}(R))$

Basis: $\text{null} \in \text{Tree}$
Recursive: If $L, R \in \text{Tree}$, and $a \in \mathbb{Z}$ then $(L, a, R) \in \text{Tree}$

Claim 1 Proof

1.

2. Base Case:

3. IH:

4. IS:

5.

Claim 2

Functions on Lists

Basis: $[\,] \in \mathrm{List}$

Recursive: If $\mathrm{L} \in \mathrm{List}$ and $a \in \mathbb{Z}$ then $a :: \mathrm{L} \in \mathrm{List}$

To prove interesting facts about lists, we need functions on lists.

Length:

Concatenation:

Claim 2

$\text{len}([\,]) = 0$
$\text{len}(a :: \text{L}) = 1 + \text{len}(\text{L})$

$\text{concat}([\,], \text{R}) = \text{R}$
$\text{concat}(a :: \text{L}, \text{R}) = a :: \text{concat}(\text{L}, \text{R})$

Claim 2: For all lists L, R, $\text{len}\big(\text{concat}(\text{L}, \text{R})\big) = \text{len}(\text{L}) + \text{len}(\text{R})$.

How do we prove a nested forall?

Claim 2 Proof

Basis: $[\,] \in \text{List}$
Recursive: If $\text{L} \in \text{List}$ and $a \in \mathbb{Z}$ then $a :: \text{L} \in \text{List}$

$\text{concat}([\,], \text{R}) = \text{R}$
$\text{concat}(a :: \text{L}, \text{R}) = a :: \text{concat}(\text{L}, \text{R})$

$\text{len}([\,]) = 0$
$\text{len}(a :: \text{L}) = 1 + \text{len}(\text{L})$

1. Let $\text{P}(\text{L})$ be "$\text{len}\big(\text{concat}(\text{L}, \text{R})\big) = \text{len}(\text{L}) + \text{len}(\text{R})$ for all lists $\text{R} \in \text{List}$". We prove $\text{P}(\text{L})$ for all lists $\text{L} \in \text{List}$ by structural induction.
2. **Base Case**:
3. IH:
4. IS:
5.