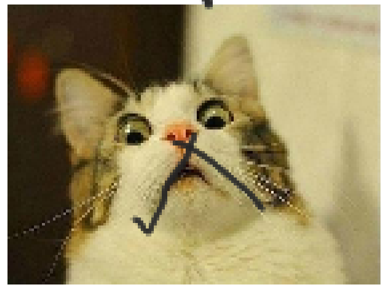
2^(n-1) guys

Leaf / Maximum: rounded up (n / 2)

Question:
Consider an arbitrary binary tree of height n. What is the maximum number of leaves that a binary tree of height n can have? What is the minimum number of leaves that a binary tree of height n can have?
Hint: Consider when the maximum amount of subtrees is achieved, and when the minimum amount of subtrees is achieved.

Scollops



height n,
height
log_2 (n-1)

Max: 2^(n-1)

2^n

h1 =

h2 =

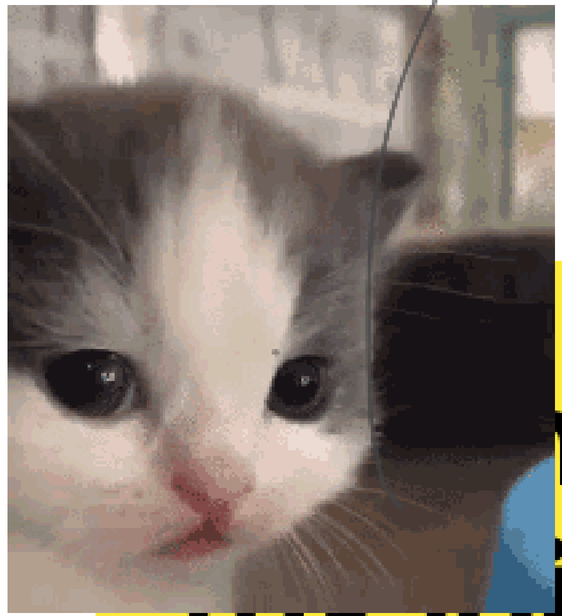
max: size n-1

min: 1 leaf, max: 2^(n-1) leaf

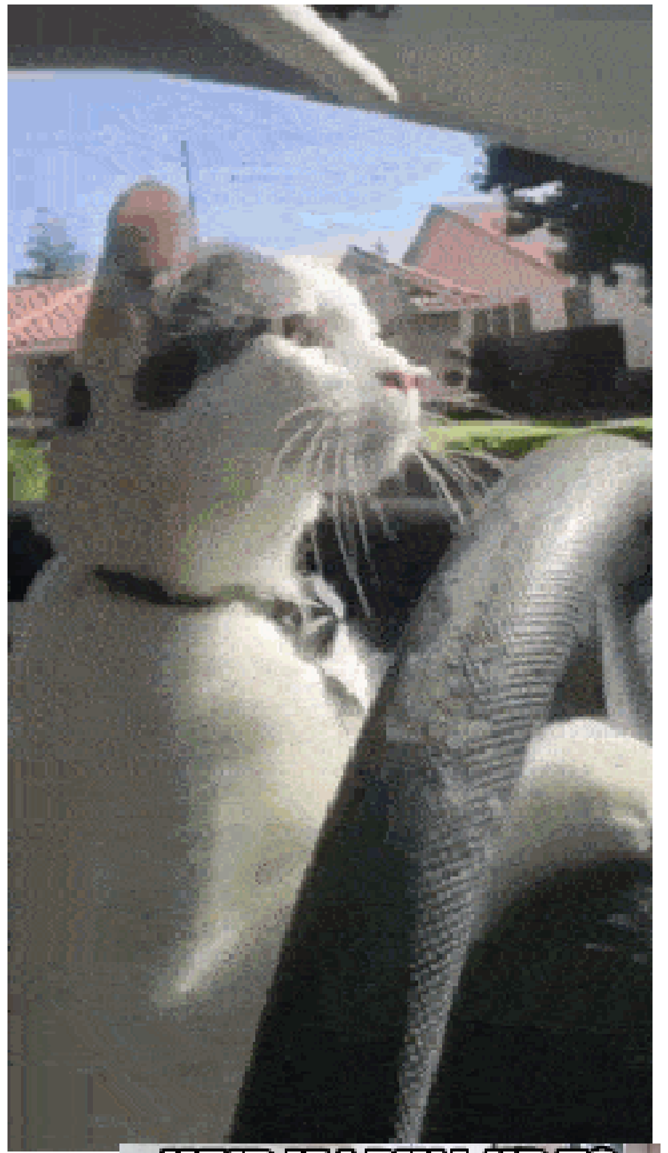
min: 1 (the root)

Harambee is alive

ibra java men shuned



n at
my lois im
still the
family guy



ibra java mentioned

Question:

Consider an arbitrary Binary Tree of size n. What is the maximum height h_1 that can be achieved? What is the minimum height h_2 that can be achieved? Justify your answer.

Hint: Consider modelling a few examples of arbitrary size n and see if you can find a pattern.

Min:
log2(-1)

Max:
n

Question:
Consider an arbitrary Binary Tree of size n. What is the maximum height h_1 that can be achieved? What is the minimum height h_2 that can be achieved? Justify your answer.
Hint: Consider modelling a few examples of arbitrary size n and see if you can find a pattern.

max height n, min size 2

ma x = n for h1

min: 2 if n != 1

Max: n / Min: floor(log_2(n)) + 1

Max height n, min height floor(log_2 (n) + 1)

max h1: n

Max: n / Min: log2(n) + 1

max: n, min: ceil(log_2(n+1))

8th slide :speaking head emoji:

Slide 8 ->

Question:

Consider the `search` operation for a *Binary Search Tree*. What is the **worst case** time complexity for the `search` operation? When does it happen? Give an example of a tree that would cause this to happen.

Are BST operations *in general* always $\mathcal{O}(\log n)$? Why or why not?

Hint: Recall our discussion about the Balancing Factor and AVL Trees earlier in the slides.

Worst case is O(n), it happens when every node only has one child and you search for the last item in the tree.

Worst case is when the element isn't even in the tree

O(n)?

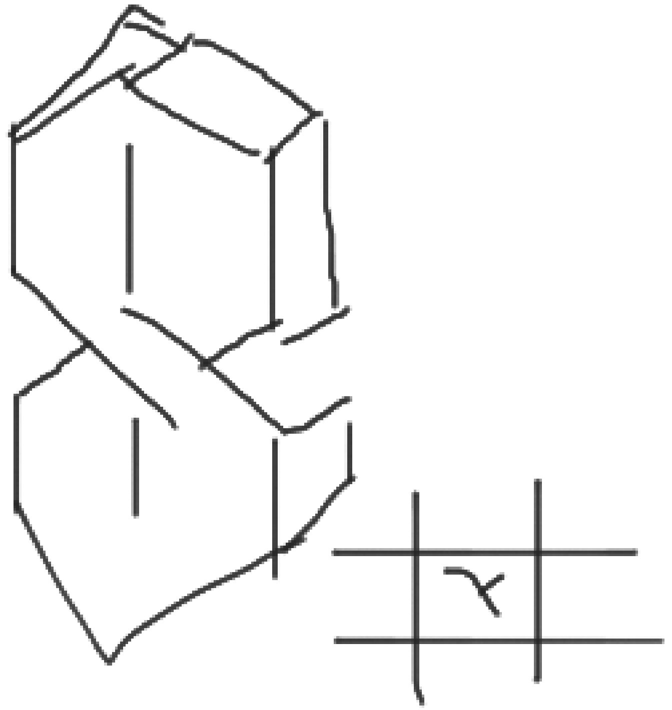
O(height)

When it's more than the max or less than min or not even found in the leaves

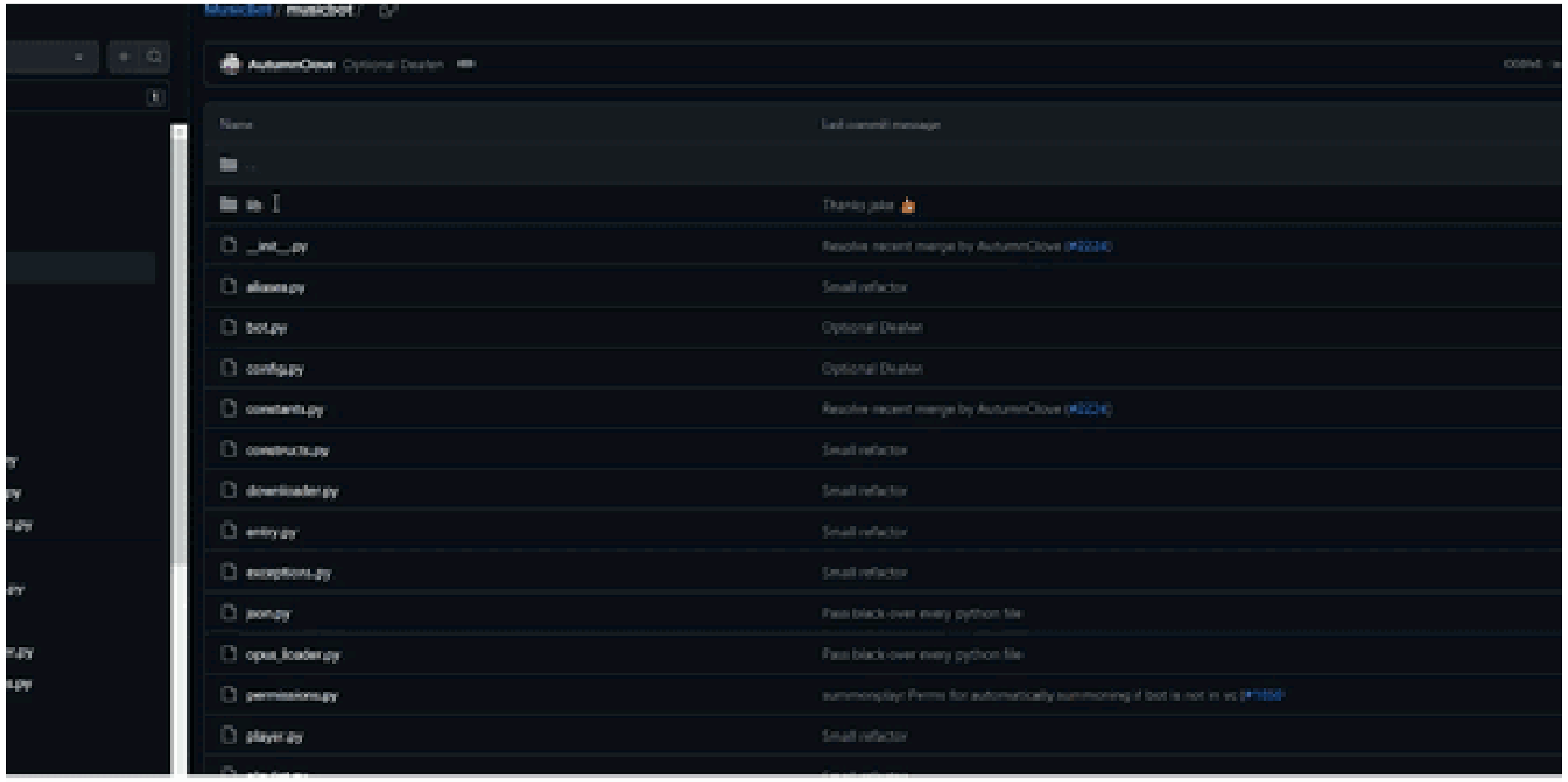
When you have a tree where each node is smaller than the rest, so a linear one, and you search for something smaller than everything in the tree.

O(n)

No because BSTs are not necessarily balanced.

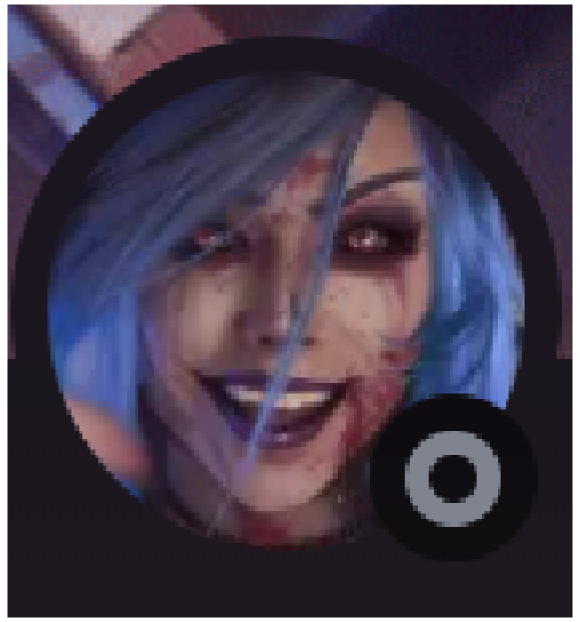


O(n)g



thanks for inspiration

easter egg



binary_tree of sam198

!?!?!?!?!?
!?!?!?!?!?

this is worrying
- Ibrahim

csc148 was here