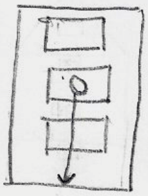
CS157 Lecture #3

EGG DROPPING PROBLEM (from Wikipedia)

* always start a proof by listing your assumptions
↳ otherwise your grader won't know what you're talking about!

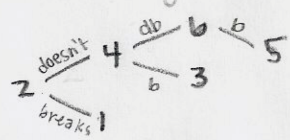


"return **last** floor **it** does not break on" ⟶ **AMBIGUOUS**

- last ↓ where do you start counting? ↓ "highest"
- it ↓ what? ↓ "a standard egg"

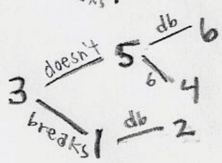
* suppose 2 eggs are available. What is the least number of egg-droppings that is guaranteed to work in all cases for a building with 6 floors?
↓
GOAL: tree that is as shallow as possible that doesn't branch left (for case when breaks) twice

idea #1: "you can only move up by 2"



= 4 egg-droppings

idea #2



= 3 egg-droppings

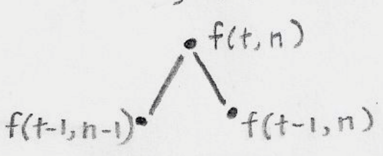
* PROOF TIP #2: notice little confusion before it spreads! If you don't know a term, stop and ask (if in class, raise your hand)

* PROOF TIP #3: emphasizing words (by underlining, capitalizing, etc) does not clear up confusion — don't do it!

* let n be #eggs, k be #floors. How can we define $W(n,k)$ (= min #egg-drops) in terms of the trees we used above? What are the rules for building such a tree?

idea #1: $W(n,k)$ is the minimum depth of a binary search tree with k nodes and where the maximum number of left moves to reach a node is n

* let $f(t,n)$ be the most floors given t trials, n eggs (i.e. at most t depth, at most n left moves, $f(t,n)$ = most # nodes to add to tree)



$\Rightarrow f(t,n) = 1 + f(t-1,n-1) + f(t-1,n)$