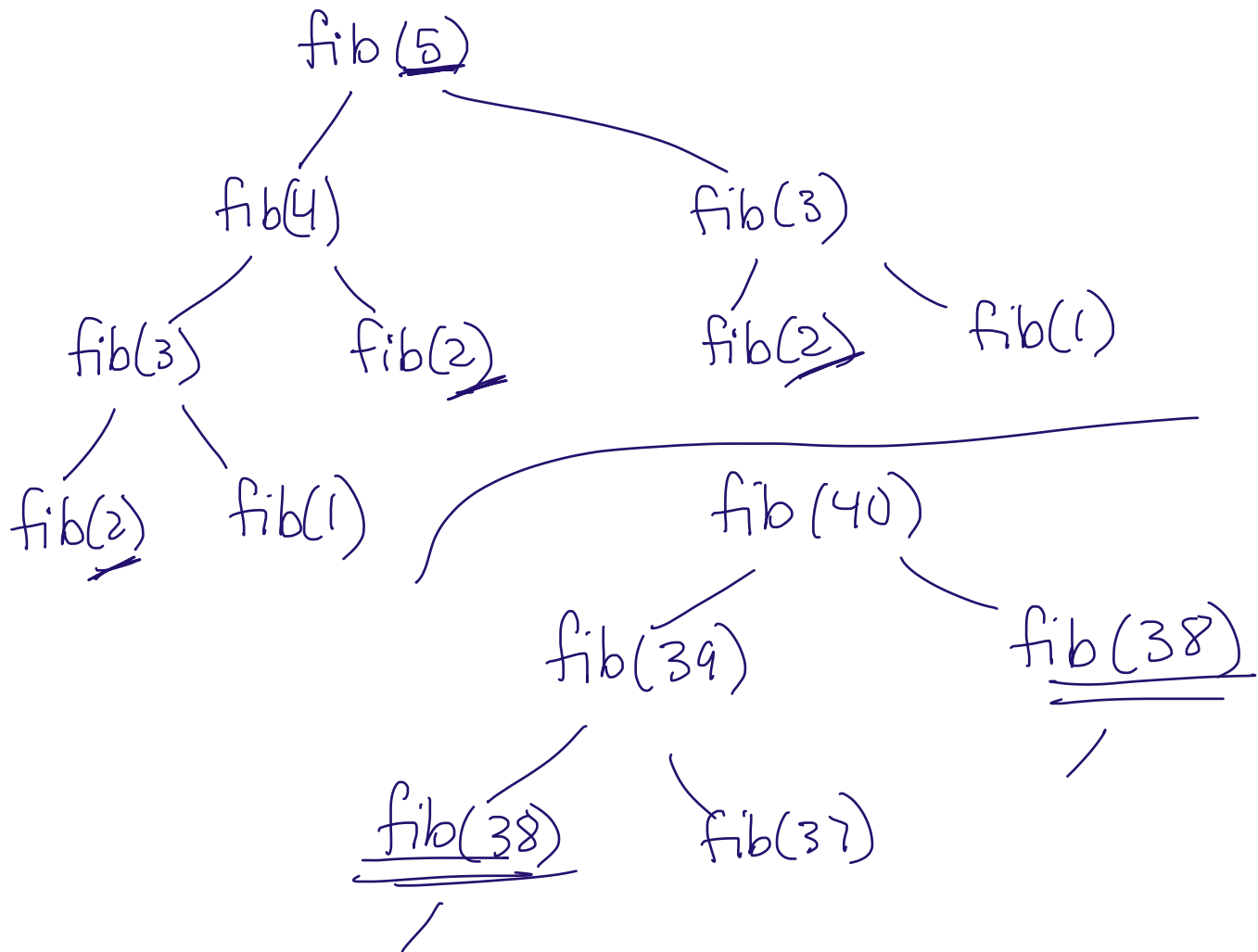


Efficiency (time)

- algorithm

Fibonacci numbers

trace out more than we're supposed to. -



Memoization

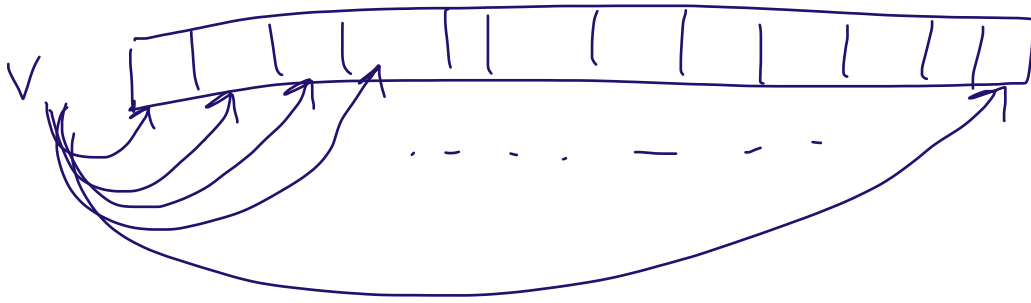
Memoization

- save computations so you can look them up later

Efficiency of --contains-- (in operator)

$v \in L$

t time to
compare v
to one item



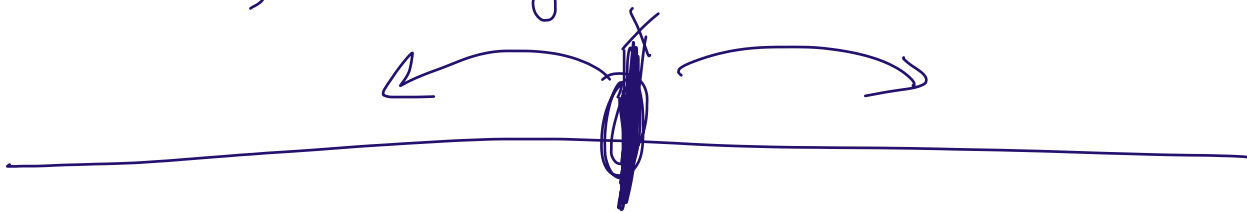
double the length

→ double the time.

doesn't matter if Python list
or linked list.

restrict to sorted lists

→ binary search.



of times we can divide by 2

until we reach 1

= # of times multi 1 by 2

to get $\geq$ # of elements n

⇒ time proportional $\log_2 \overbrace{\text{\# of elements}}^n$

$\log_2 = \lg$ $\lg(10^3) \approx 10$

list with	time for linear	time for binary
10^6	$\sim 10^6$	~ 20
10^9	$\sim 10^9$	~ 30
10^{12}	$\sim 10^{12}$	~ 40
10^1	~ 10	$\sim 3 \text{ or } 4$

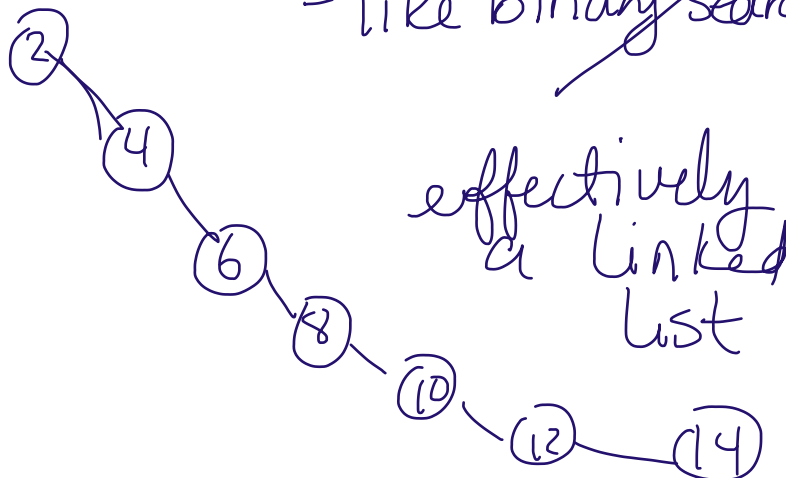
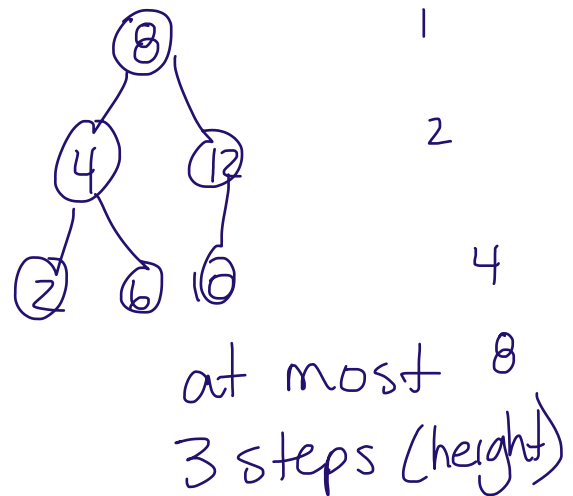
contains on trees?

- general tree
 - time proportional to # of nodes

- binary tree
 - same

- binary search tree

- better.
 - like ~~binary search~~



effectively
a linked
list

"balanced"
 - each level
 (except bottom)
 is full $\Rightarrow$ height
 is minimized