Homeworks are now due every Tuesday at 9pm

Splay trees

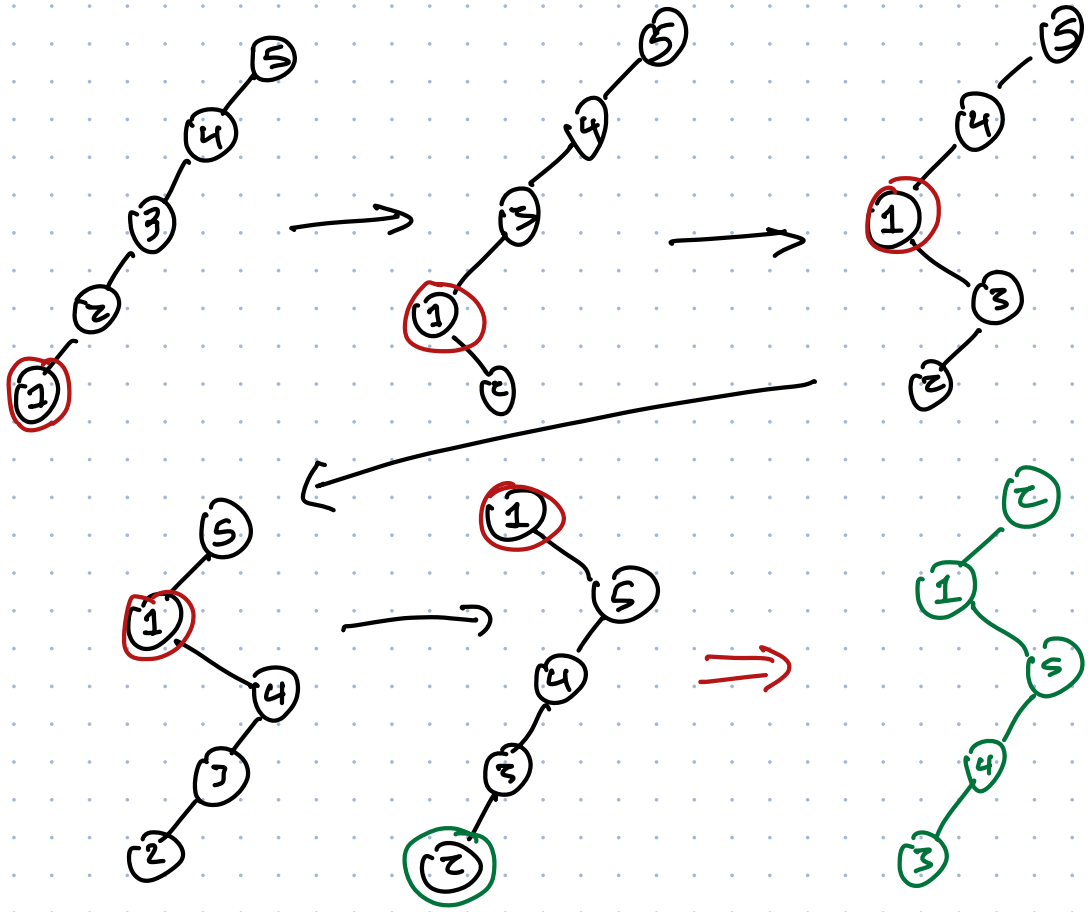
Intuition: self-adjusting

move the last search key to root using rotations

Idea: Find(x): rotate at x until x is root

Insert(x):

BAD IDEA

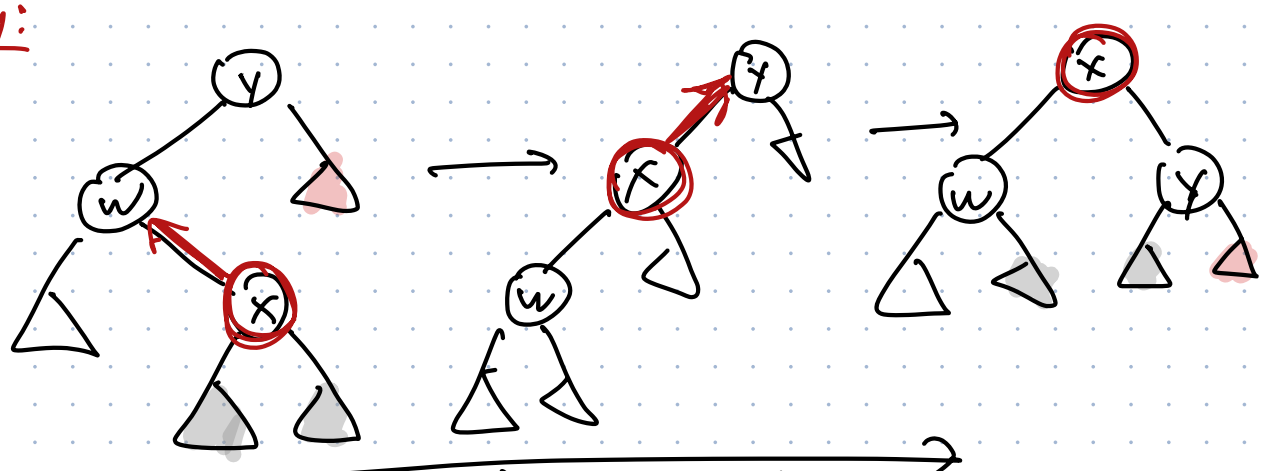


N searches takes $\Omega(N_n)$ time

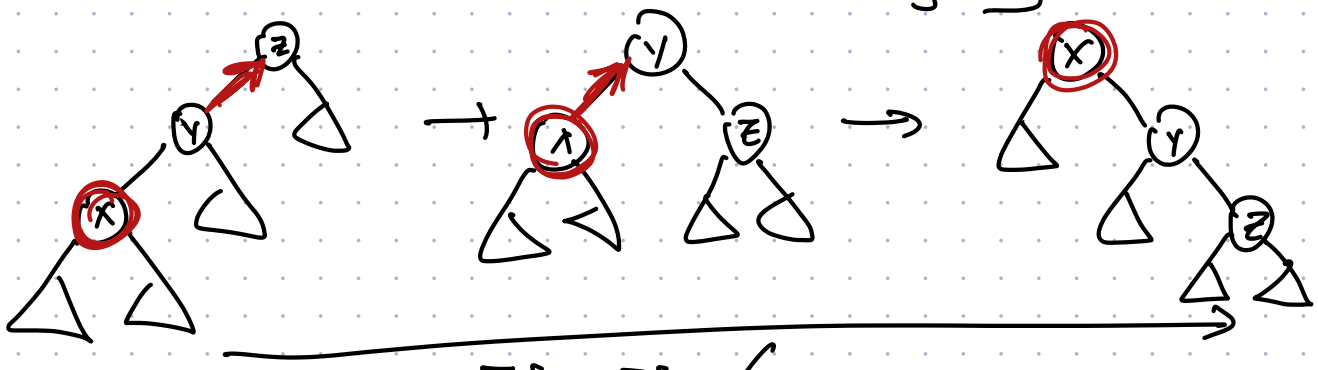
$\Rightarrow \Omega(n)$ time each

a_m

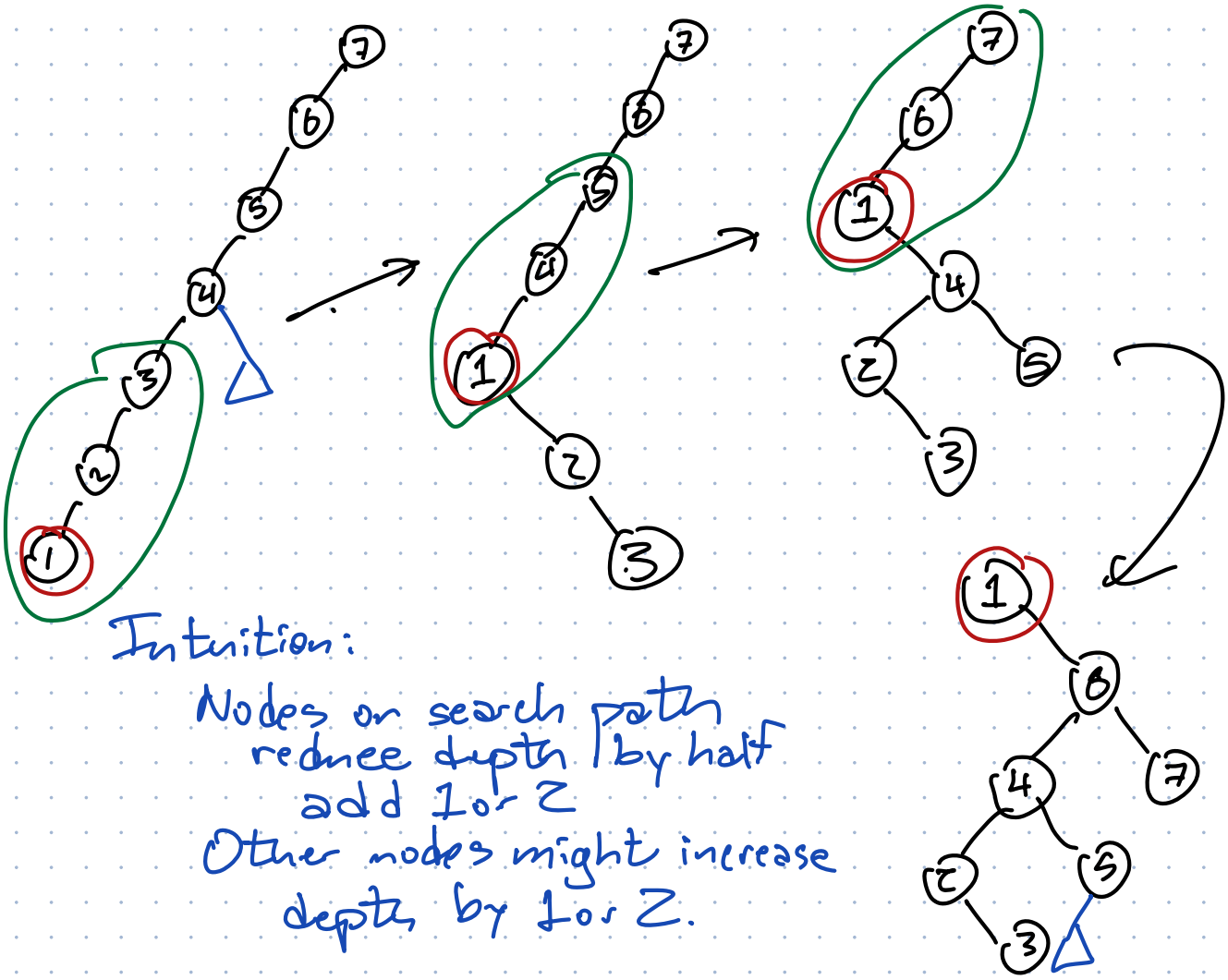
Splay:



zig-zag / zag-zig



zig-zig / zag-zag
roller coaster



Intuition:

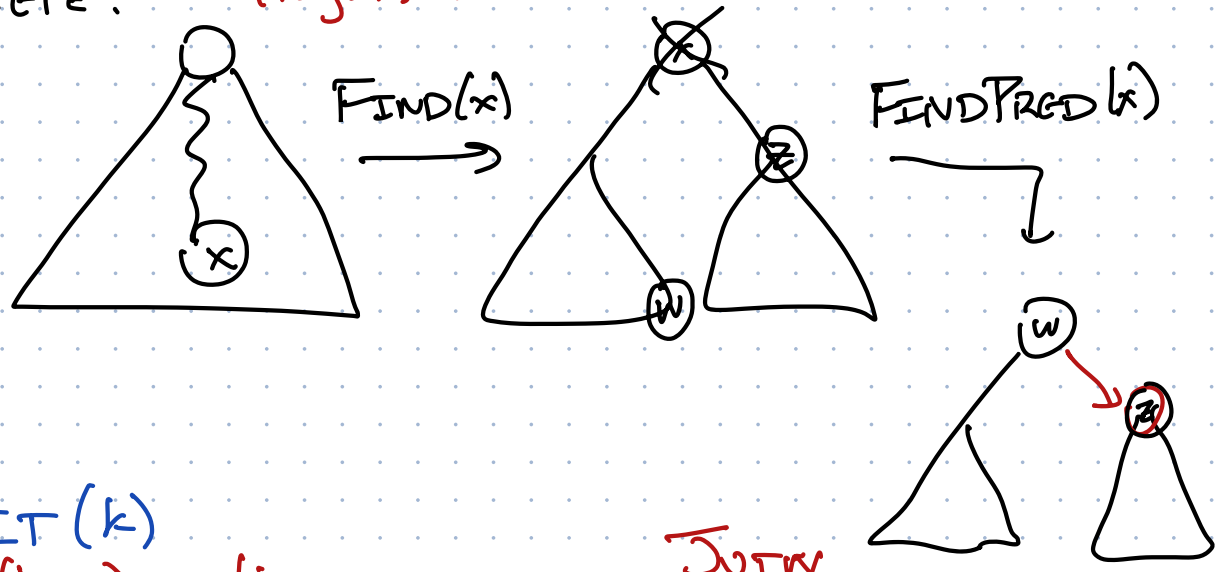
Nodes on search path
reduce depth by half
add 1 or 2

Other nodes might increase
depth by 1 or 2.

Amortized # rotations to perform one splay
$\leq 1 + 3\log_2 n$ [MAGIC!]

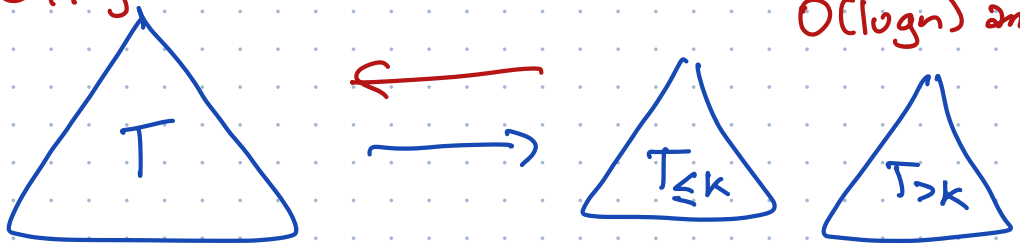
$\Rightarrow$ Am. time for FIND $= O(\log n)$
INSERT $= O(\log n)$

DELETE? $O(\log n)$ am. time



SPLIT(k)
$O(\log n)$ am. time

JOIN
$O(\log n)$ am. time



Rope: store set of strings
support operations:

$O(1)$ NewString(a) — new string of length 1

$O(\log n)$ am. time:
- Concat(S,T) — replace S and T with ST
- Lookup(S,k) — return S[k]
- Split(S,k) — replace S with S[1..k] S[k+1..|s|]

Text editors

Each string in a splay tree

SPLAYTREE →

MAGIC:

- $O(\log n)$ am. time
- If each node x is accessed $t(x)$ times
  $O(\log \frac{T}{t(x)})$ where T = total # accesses
  Static optimality
- Pick favorite node F "Finger"
  $O(\log \text{dist}(F, x))$
- D = # distinct items accessed since last access to x
  $O(\log D)$

Conjecture:
Within $O(1)$ factor of best dynamic BST