

II Motivation : can we just do "one sweep (top-down)"

Optimizing Find(e) $\rightarrow$ $O(h)$

Find(t)

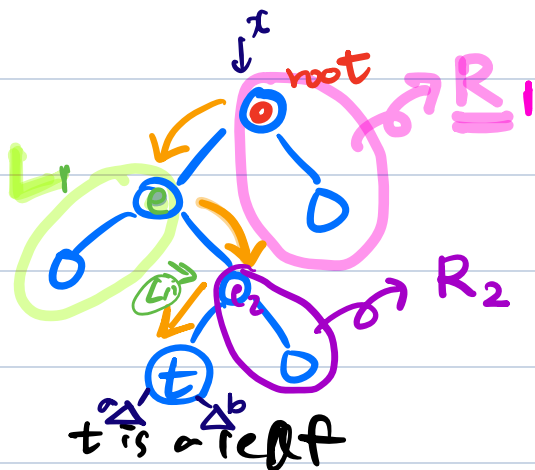
both

BST search of e

bringing e all the way to the top

Basic Idea

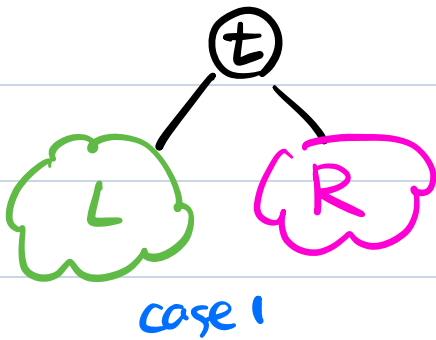
As we go down the tree searching for target t we move the nodes that are on this search path to either L, R, where L and R are the eventual left and right subtree of t.



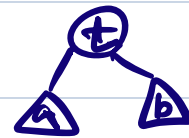
$$R_1 \gg R_2$$

$$L_1 \ll L_2$$

t is not a leaf

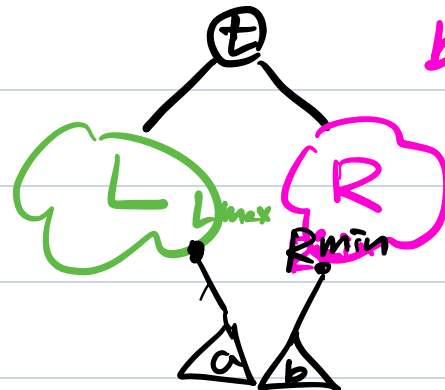


case 1



$$b \ll R_{\min} \ll R_{\max}$$

$$a \gg L_{\max}$$



case 2

3 Detailed Algo

yet to be traversed

Initially, $L = \emptyset$, $R = \emptyset$, $\underline{M} = T$, $x = \text{root}$
entire tree

at any given time

always maintain $L_{\max}$, $R_{\min}$

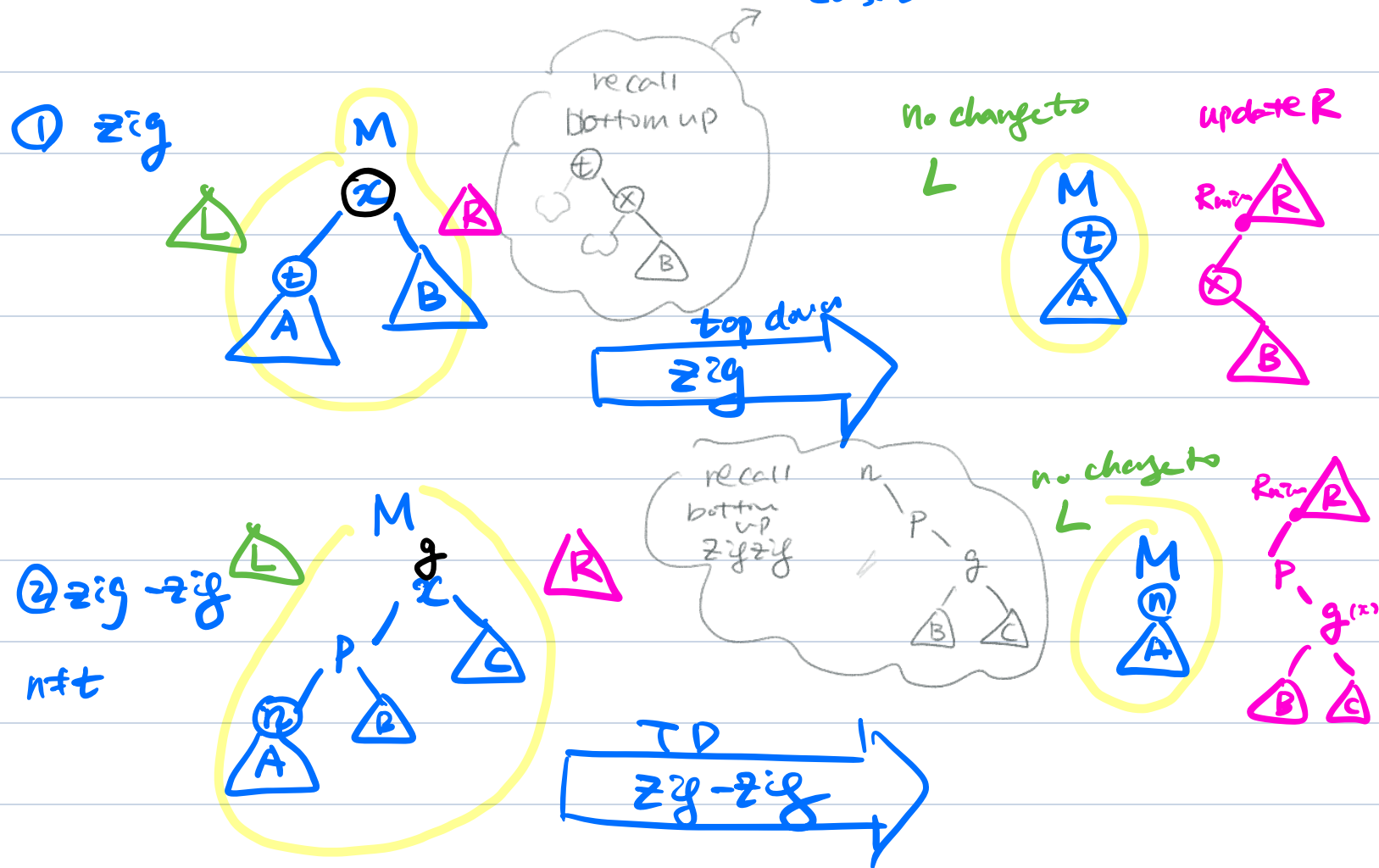
$t = \text{target}$

As we go down the search path for target t

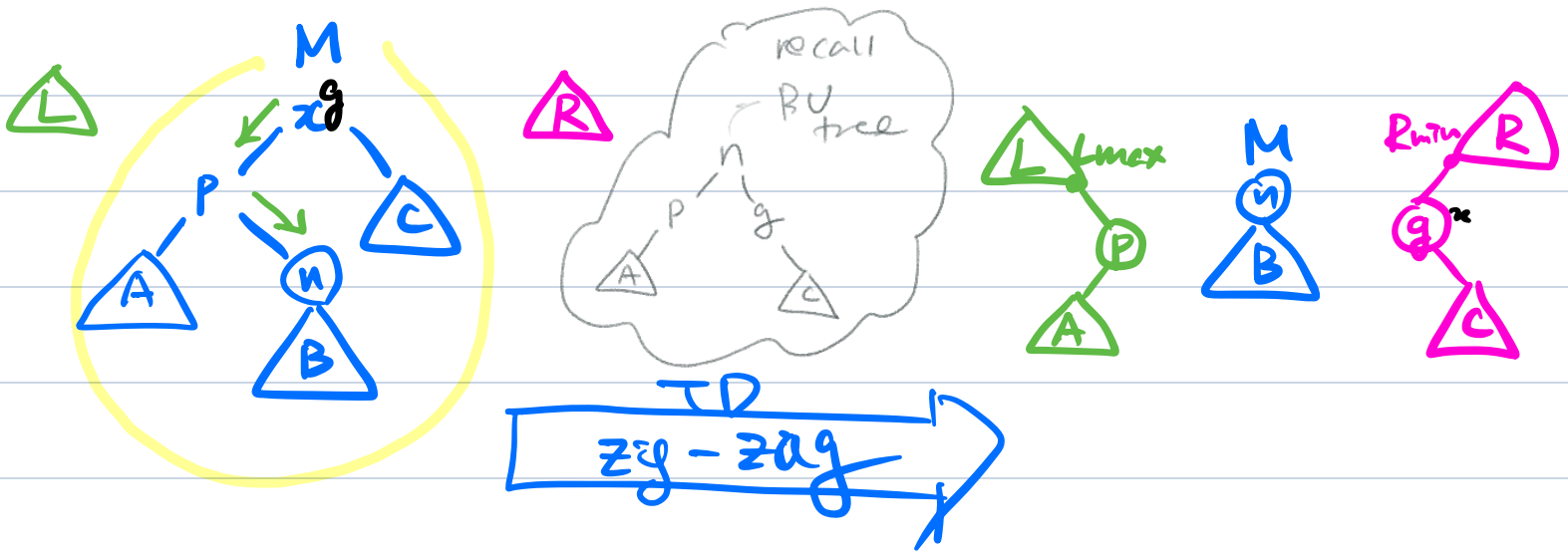
update $L, R, M + L_{\max}, R_{\min} + x$

until M 's root is target.

* Met $\rightarrow$ Attach $\begin{cases} \text{case 1} \\ \text{case 2} \end{cases}$ as shown above



③ zig-zag



p_0 above 1, 2, 3 until n is target
 or n 's child is the target