CSE 101 12-3-20

Pa6: ext. 2 days (last).

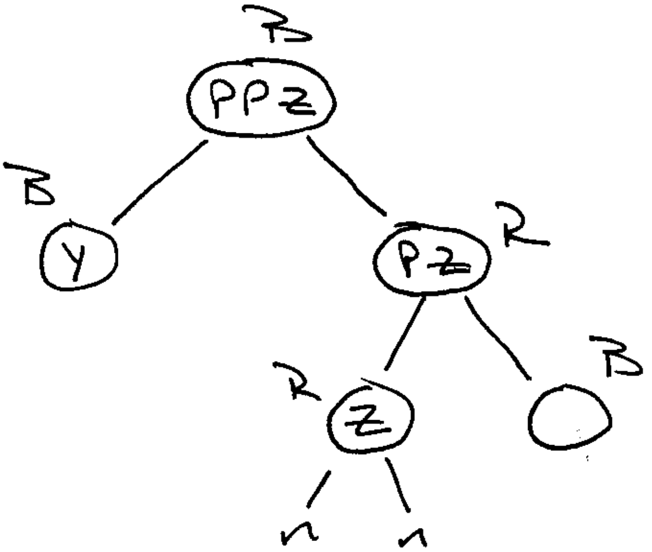
Pa7: ext. 2 days.

Pa6 stuff: analogies

List	Dictionary
moveFront	beginForward
" Back	" Reverse
get	current Key " Val
moveNext	next
" Prev	prev

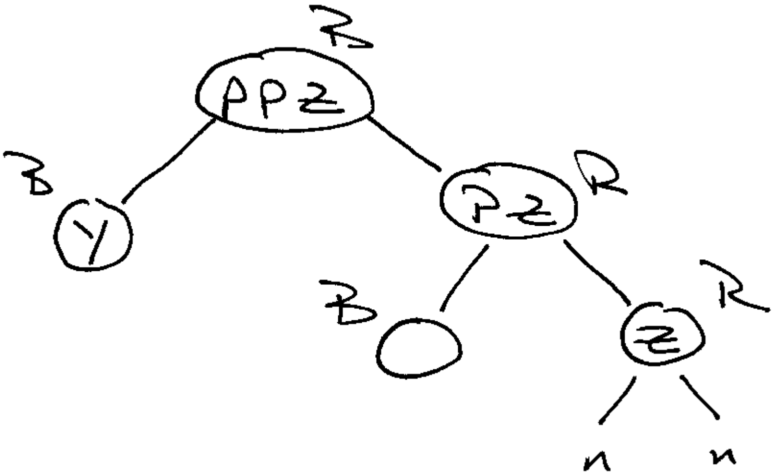
continue RB-Insert()
RB-InsertFixUp()

case 5: y black, z is Parent's left child



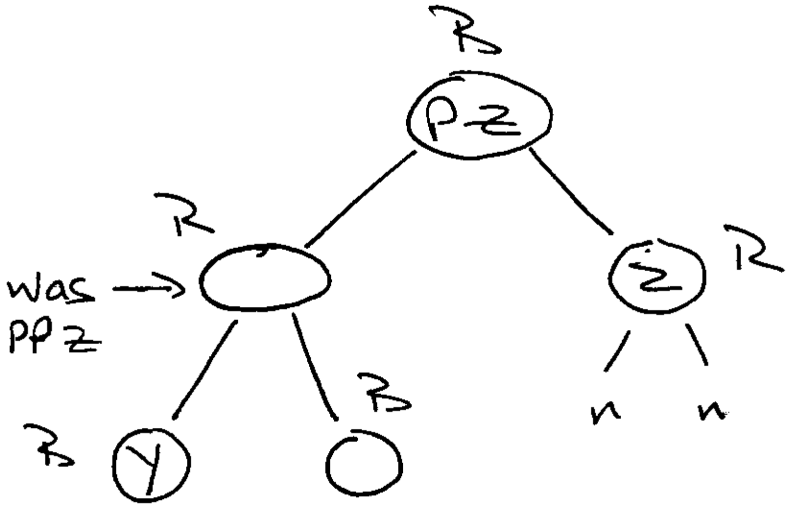
convert to case 6. let z
climb up to Pz, RightRotate(T, z)

case 6: y black, z is right child of Pz.



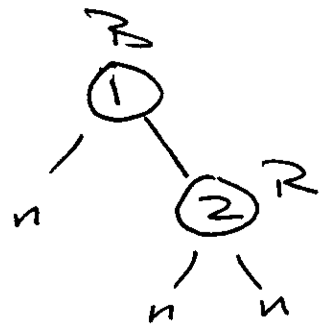
color PZ black, color PPZ Red

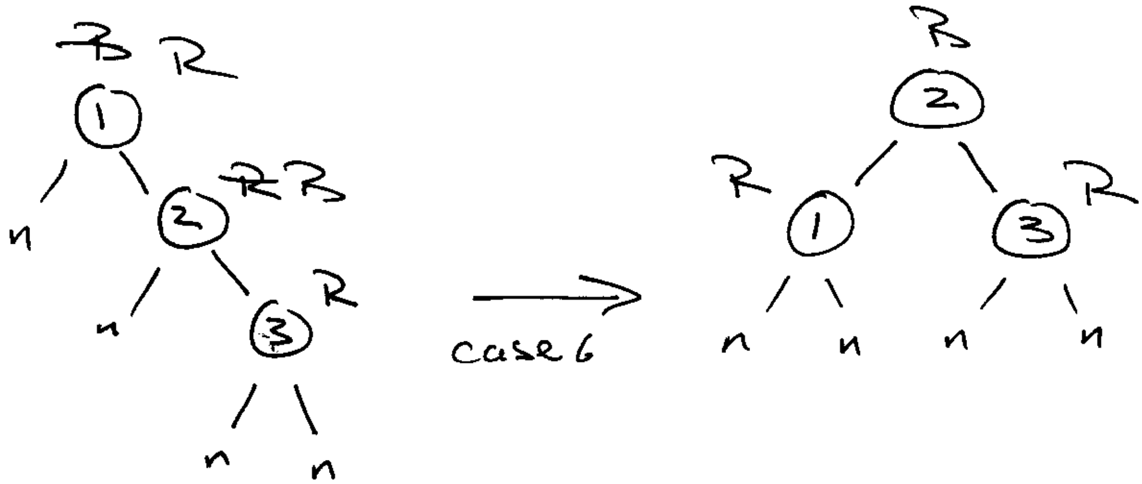
then LeftRotate(T, PPZ)

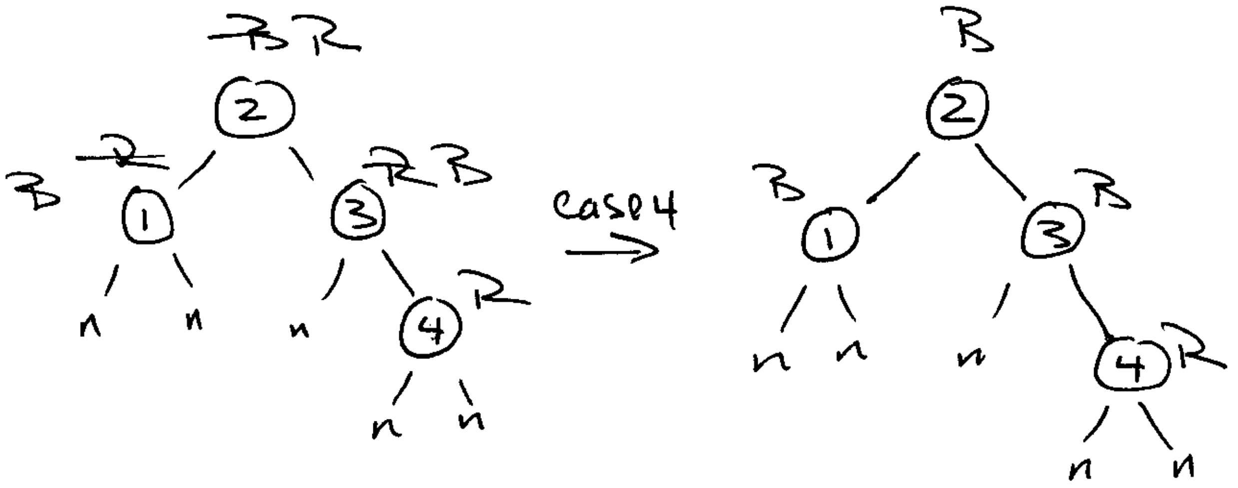


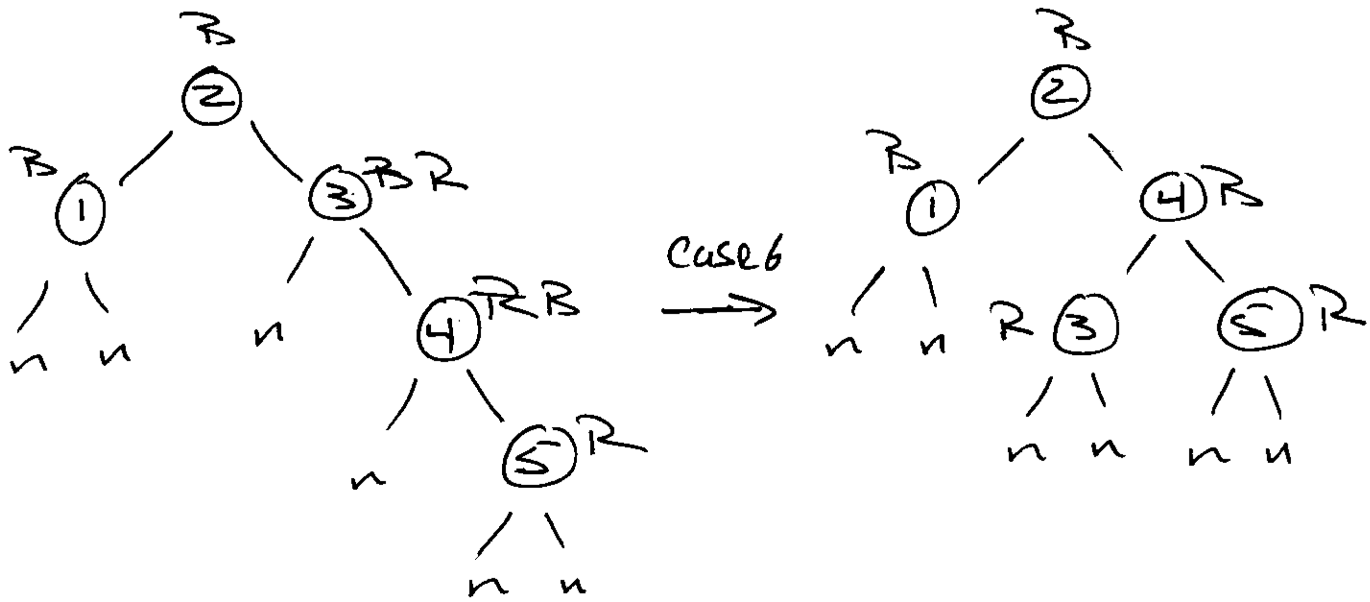
note: RBT Prop 4 & 5 are now satisfied. also height(T) could have been lowered. At this point, while loop ends.

Ex. insert: 1, 2, 3, 4, 5 in an initially empty RBT.






case4


Case6

Chapter 24

SSSP Problem in weighted graphs.

Defn

a weighted (digraph) is a digraph $G = (V, E)$ with a function

$$w : E \to \mathbb{R}$$

let $P : v_0, v_1, v_2, \ldots, v_k$ be a $v_0 - v_k$ path in G. The weight of P is

$$w(P) = \sum_{i=1}^{k} w(v_{i-1}, v_i)$$

let $x, y \in V(G)$. the <u>shortest path weight</u> $\delta(x,y)$ (or <u>minimum path weight</u>) is

$$\delta(x,y) = \begin{cases} \min\{w(P) \mid P \text{ is an } x\text{-}y \text{ path}\} \\ \quad \text{if } y \text{ is reachable fr. } x \\ \infty \quad \text{otherwise} \end{cases}$$

a <u>shortest path</u> P is an x-y path s.t. $w(P) = \delta(x,y)$.

<u>SSSP Problem</u>

given $s \in V(G)$, determine

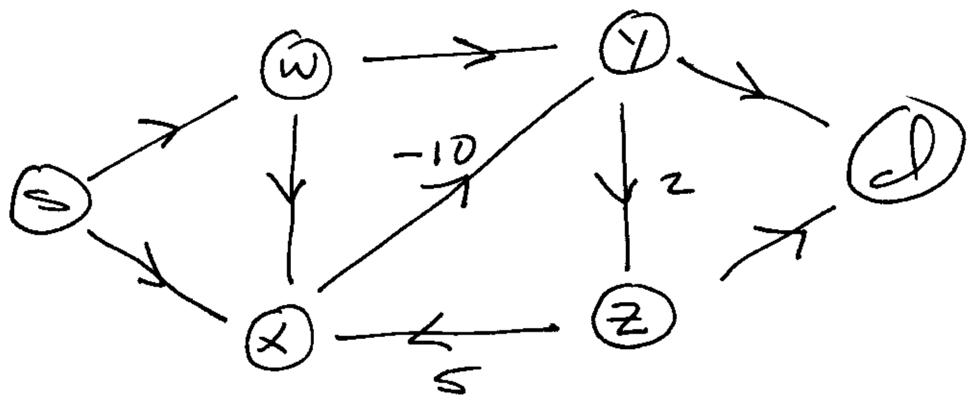
(1) $\delta(s,x)$ for all $x \in V(G)$

(2) for all x with $\delta(s,x) < \infty$, find a shortest s-x path.

Two algorithms

- BellmanFord: allows negative edge weights
- Dijkstra's: Requires non-negative edge weights.

Ex



observe there is no min-weight s-d walk.

Problem: neg. weight cycle reachable from source.