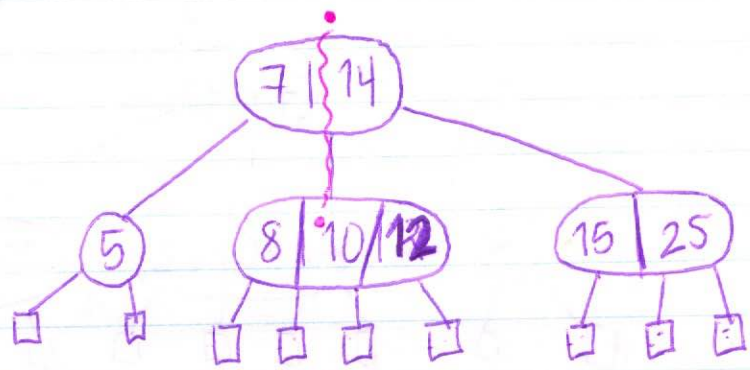
Nov 15
COMP 2402

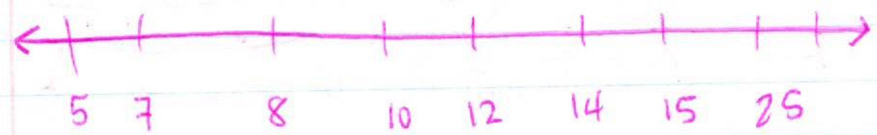
2-4 Trees



Degree: every internal node has 2-4 children

Height: every leaf has the same depth

Odering: u.key[i-1] < u.child[i] < u.key[i]



n keys, n+1 intervals (leaves)

if h: then min # of leaves $= 2^n$
max # of leaves $= 4^n$

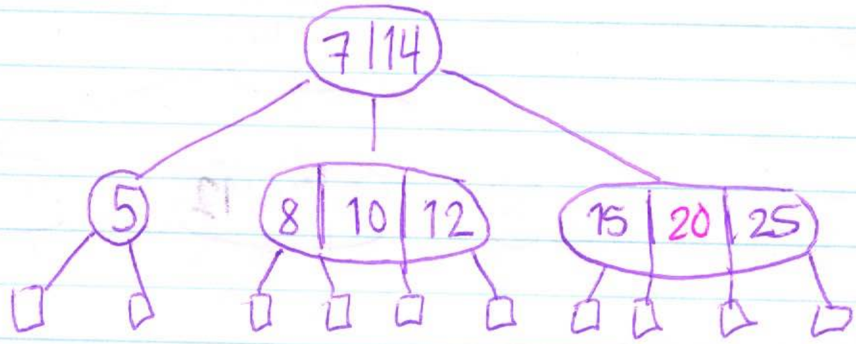
$2^h \leq \#\text{ of leaves} \leq 4^h$

$\log_2(2)^n \leq \cancel{n+1}\ \log_2(n+1)$

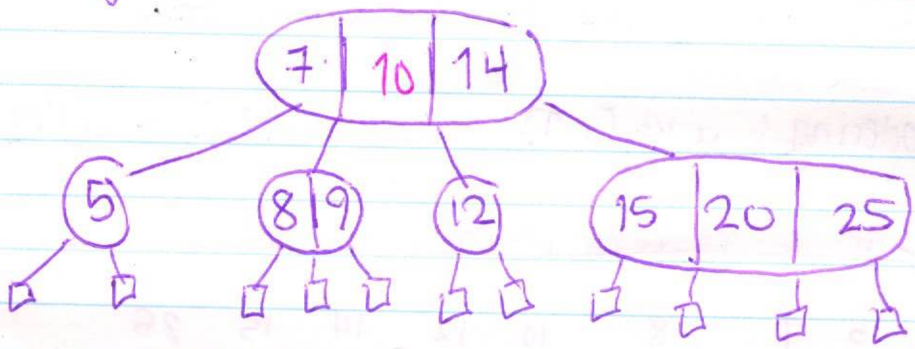
$n \leq \log(n+1)$

ADD

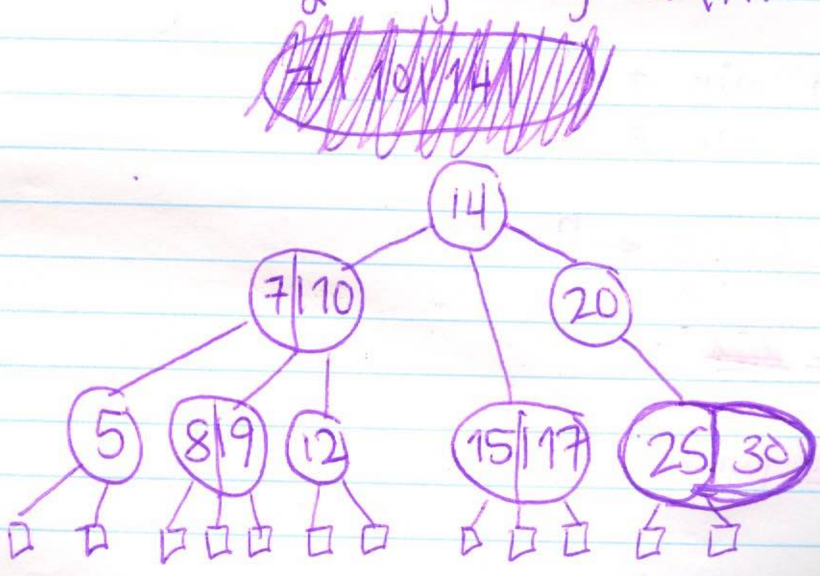
case 1: u has < 4 children (eg add(20))



case 2: u has 4 children (eg add(9))
(overflow)



case 3: cascading overflows (eg add(17))



$= O(\log(n))$

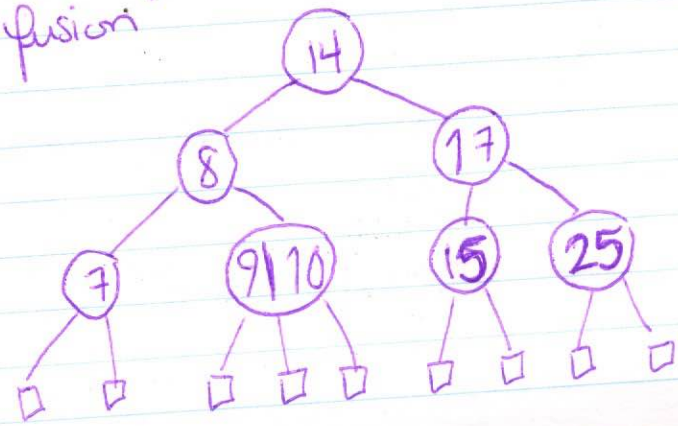
REMOVE

case 0: u is a leaf with more > 2 children (eg. remove(30))

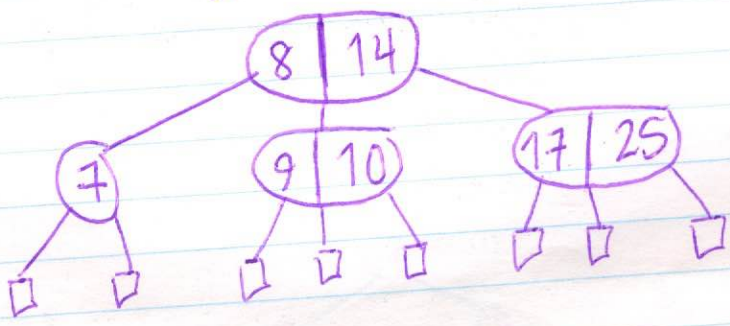
case 1: internal node (eg remove(20))
↳ swap with predecessor & cut

case 2: underflow with 3-4 sibling (eg remove(5))
↳ transfer

case 3: underflow without 3-4 sibling (eg remove(12))
↳ fusion



case 4: cacasiding unflow (eg remove(15))



= O(log(n))

Red Black Trees

Root : The root is black

External : All external nodes are black

Red : the children of a red node are black

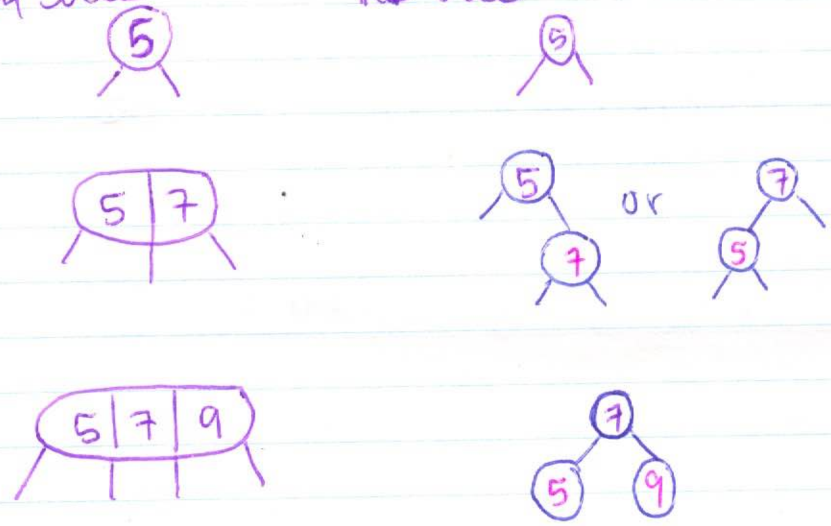
Black : All leaves have the same "black-depth"

2-4 tree

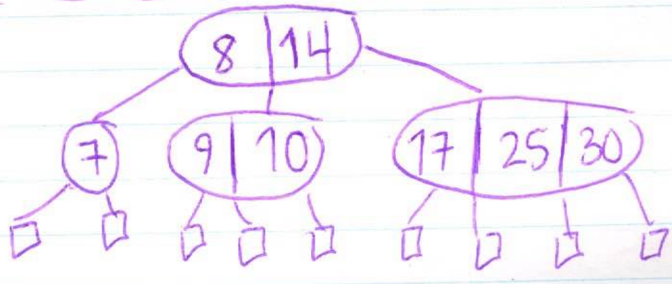
~~RB~~ tree

= black

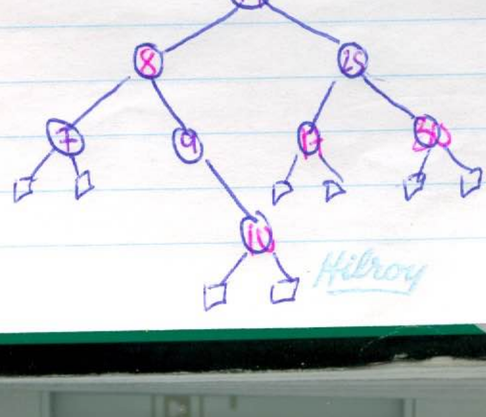
= red



2-4 tree



RB tree



O(log(n)) black depth