COP 5536
Advanced Data Structures

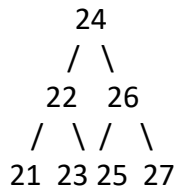
Spring 2018

Exam 2 Solutions

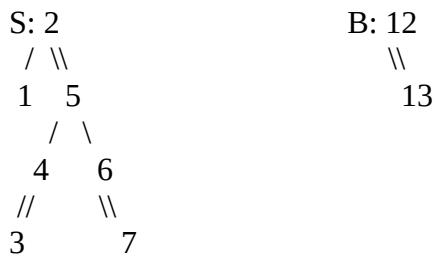
Question.1 (12):

Start with an empty AVL tree and perform insert operations using the following sequence of keys: 21, 23, 27, 26, 24, 25, and 22. Show each step.

rotations : RR (3 points) -> LL (3 points) -> RL (3 points) -> LR (3 points)

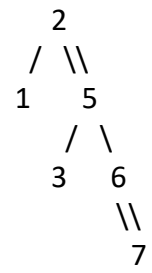


Question 2 (12):
For the following red-black trees (double links are red links),



(a) (6) Consider the red-black tree above. Perform delete (4) operation for red-black tree S, showing each step.
(b) (6) Perform Join (S; 10; B) on the **ORIGINAL** red-black trees in the above figure, showing each step.

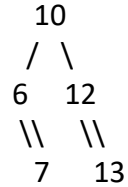
(a) After Delete 4 (6 points – Marks are given for partially correct answer)



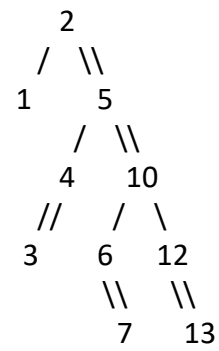
(b) Numbers in parentheses represent ranks.

i: Follow the right child pointer until rank(B) = rank(x), where rank(B) =1, x is a node pointer of tree S.

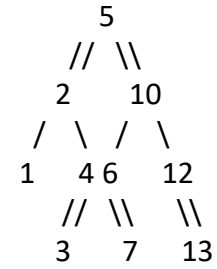
ii: Combine subtree x, 10, and B (1 Point)



iii: Combine the result of ii to node 5 through a red pointer (3 Points)

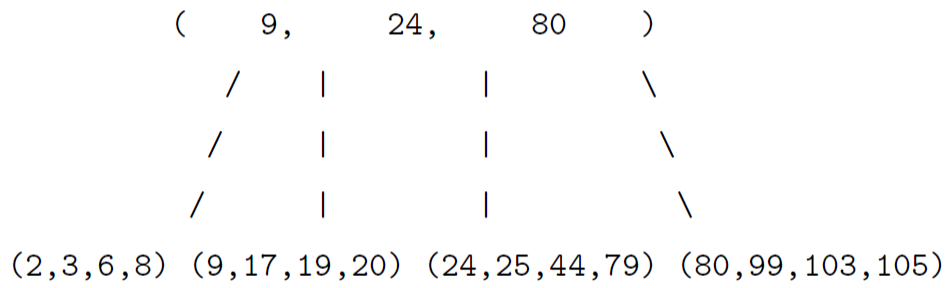


iv: Perform a RR rotation (2 Points)



Question 3 (12):
(a) (6) Suppose that n keys are inserted into an empty B-tree of order m. What is the maximum height h in terms of n and m in the final B-tree? Show how you derive your answer.

(b) (6) Given the following 5-way B+-tree, insert 40 and 100 in this order. Show each step and the resulting tree.



(a) formulated problem correctly based on correct B-tree defition : 3
Correct final answer and derivation: 3

Solution: (a) $1 + log_{\lceil m/2 \rceil}(n+1)/2$

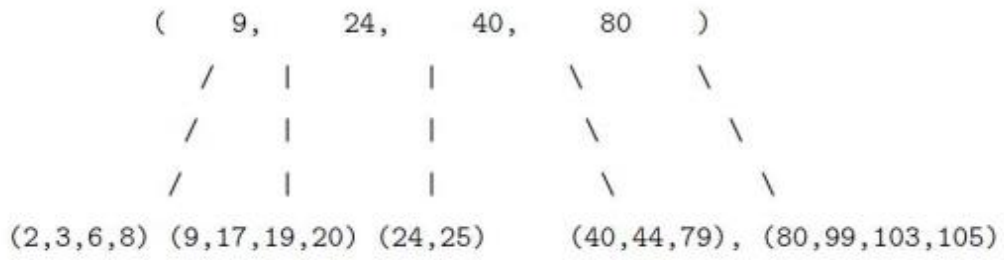
Derivation :

$n + 1 = 2\lceil m/2 \rceil^{h-1}$ (since, number of keys + 1 = total number of external nodes)

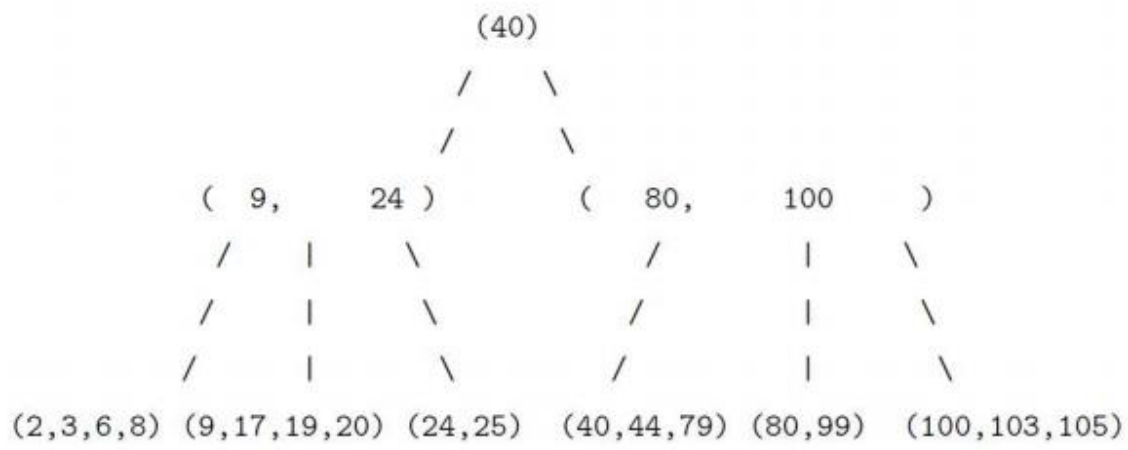
Hence, $h = log_{\lceil m/2 \rceil}(n+1)/2 + 1|$

(b) for each correct insertion : 3 points
for each minor mistake -1
if used incorrect insert algorithm : no point

insert 40



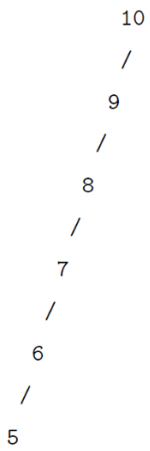
Insert 100



Question 4 (14):

(a) (7) Insert the following keys into an initially empty instance of a splay tree, assuming that this is a *bottom-up* splay tree:
4, 1, 5, 9, 3, 8, 2

(b) (7) Consider the following *top-down* splay tree:

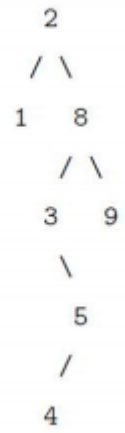


Perform a *split* operation with respect to the node with the key 5, showing each step

(a)

did not splay for each insert : 1 or no points
perform correct splay operations for each insert and arrive at correct solution : 7
each splay error : -2

Solution: (a)



(b) if knows the tree has to be divided in two parts : 2
performed splay at a correct point in a correct way : 3
splay operations has no mistake : 2

S = empty