Data Structures and Algorithms (NTU, Spring 2015)

Homework #6

6.1 Skip List, Binary Search Tree

(1) C-9.15 median() in skip lists

Indexable skiplist: For every link, also store the width of the link. The width is defined as the number of bottom layer links being traversed by each of the higher layer "express lane" links.

function median()

node ← head

i ← n/2 + 1 # don't count the head as a step

for level from top to bottom do

while i ≥ node.width[level] do # if next step is not too far

i ← i - node.width[level] # subtract the current width

node ← node.next[level] # traverse forward at the current level

repeat

repeat

return node.value

end function

(2) R-10.5 binary search tree

Insert order: 1-2-3-4 , tree : 1 ( ,2 (,3 (,4)))

Insert order: 3-1-2-4, tree: 3 (1(,2),4)

(3) C-10.12 Show that the nodes that become unbalanced in an AVL tree during an insert operation may be nonconsecutive on the path from the newly inserted node to the root.

6.2 Balanced Binary Search Trees

Hw6\_2.c

6.3 Disjoint Set

(1)

(2)

(3)

(4)

(5)