**Mason Shepherd**

**Homework**

R-11.5 [5 pts] Dr. Amongus claims that the order in which a fixed set of entries is inserted into an AVL tree does not matter—the same AVL tree results every time. Give a small example that proves he is wrong.

Ans:

(1, 2, 3, 4, 5) vs (3, 4, 1, 2, 5)

2 3

/ \ / \

1 4 1 4

/ \ \ \

3 5 2 5

R-11.8 [5 pts] Draw the AVL tree resulting from the insertion of an entry with key 52 into the AVL tree of Figure 11.13b.

Ans:

62

/ \

50 78

/ \ \

44 52 88

/ / \

17 48 54

R-11.9 [5 pts] Draw the AVL tree resulting from the removal of the entry with key 62 from the AVL tree of Figure 11.13b.

Ans:

(not including 52?)

54

/ \

44 78

/ \ \

17 50 88

/

48

R-11.14 [5 pts] What does a splay tree look like if its entries are accessed in increasing order by their keys?

Ans:

“Diagonally” shaped. Sorted or Unsorted?

R-11.18 [5 pts] An alternative way of performing a split at a node *w* in a (2,4) tree is to partition *w* into *w*′ and *w*′′, with *w*′ being a 2-node and *w*′′ a 3-node. Which of the keys *k*1, *k*2, *k*3, or *k*4 do we store at *w*’s parent? Why?

Ans:

k3 because there needs to be a key added to *w*’s parent node such that *w’* can be split to the left of the key (less than) and *w”* can be split to the right of the key (greater than).

Supposed to be k2 for some reason?

R-11.19 [5 pts] Dr. Amongus claims that a (2,4) tree storing a set of entries will always have the same structure, regardless of the order in which the entries are inserted. Show that he is wrong.

Ans:

(1, 2, 3, 4, 5) vs (4, 2, 5, 1, 3)

(3) (4)

/ \ / \

(1, 2) (4,5) (1, 2, 3) (5)

R-11.23 [10 pts] For the following statements about red-black trees, provide a justification for each true statement and a counterexample for each false one.

a. A subtree of a red-black tree is itself a red-black tree.

Ans:

False. The root of the subtree could be “red” preventing it from being a proper red-black.

b. The sibling of an external node is either external or it is red.

Ans:

True. In any instance that the sibling is not external or red, there would be too many black nodes in a row to uphold the depth property.

c. There is a unique (2,4) tree associated with a given red-black tree.

Ans:

True. If for every black node you group it with any of its red children (if there are any), you will get the proper nodes for the corresponding (2, 4) tree.

d. There is a unique red-black tree associated with a given (2,4) tree.

Ans:

False. The reverse of the aforementioned process can’t be applied to all (2, 4) trees.

C-11.29 [10 pts] Explain how to use an AVL tree or a red-black tree to sort *n* comparable

elements in *O*(*n* log *n*) time in the worst case.

Ans:

W/ an AVL tree:

1) enter the set of elements into an empty AVL tree O(*n*) time

2) use firstEntry() or lastEntry() to reacquire each element in the desired order. O(log*n*) time

step 2 will be done *n* times therefore this should be O(*n*log*n*) time.