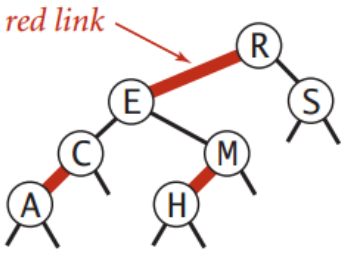
COS 226 – Data Structures and Algorithms
Fall 2014 – Flipped Lecture Section
Individual/small group worksheet - SOLUTIONS
Week 6 – 10.04.14
Topics covered: LLRB, hashing

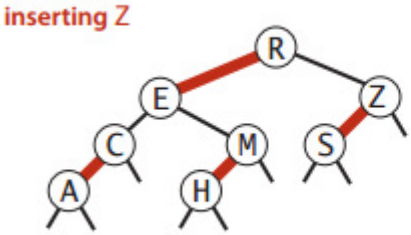
Instructions: This worksheet covers 2-3 trees, LLRBs and hashing. Read the worksheet first (before viewing the videos) and understand what type of questions needs to be answered. As you watch videos, if you find the answer to a problem, write the answer here and if possible in salon, so you can share it with others. Also be sure to make some comments/questions on salon.

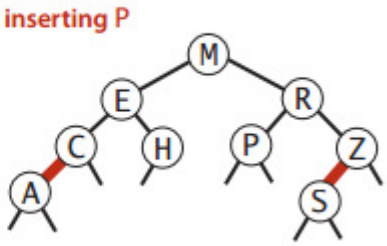
1. LLRB - Consider the following LLRB.



(a) add key Z and then add key P and show the resulting tree after insertions

ANSWER:

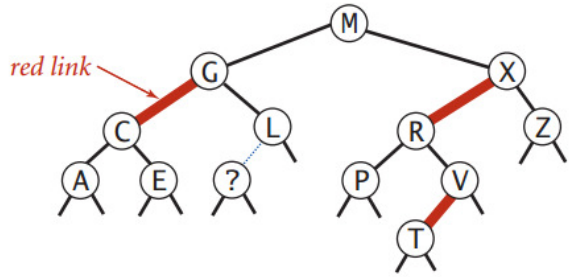




(b) How many total left rotations, right rotations and color flips are performed to insert these two keys?

2 left rotation (one while inserting Z, one while inserting P), 1 right rotations (while inserting P), 2 color flips (while inserting P).

2. Consider the following LLRB

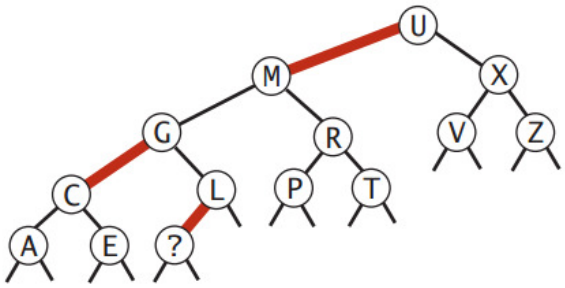


(a) Which one of the keys below can be in the node marked?

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

ANSWER: H I J K

(b) Add the key U to the LLRB above and draw the tree



3. Insert the keys E A S Y Q U E S T I O N in that order into an initially empty table of M=5 lists using separate chaining. Use the hash function 11k % M to transform k^{th} letter of the alphabet to a number.

E(5) A(1) S(19) Y(25) Q(17) U(21) E(5) S(19) T(20) I(9) O(15) N(14)

0 ➔ E Y T O
1 ➔ A U
2 ➔ Q
3
4 ➔ S I N

4. Suppose that 10,000 strings of length 5 (all strings are formed using alpha characters) are inserted into a linear probing hash table (of size = 10000) using the function, hash (key) = sum of the characters in the key (note that each key is 5 characters long).
   a. What is the load factor of the hash table?
   b. What is the probability of a collision after 125 keys are inserted into the table?

Without loss of generality, we can assume that A=0, B=1, etc.
Therefore the hash code for any string of length 5 would be between [0…125]
(a) Load factor = 10000/126 ~ 8.0
(b) Probability of a collision – (10000-126)/10000

5. Match up the following algorithms with function on right. You can use a letter more than once or not at all.

___ *Min* height of a binary heap with N keys.	A. ~ 1
___ *Max* height of a binary heap with N keys.	B. $\sim \frac{1}{2} \lg N$
___ *Min* height of a 2-3 tree with N keys.	C. $\sim \log_3 N$
___ *Max* height of a 2-3 tree with N keys.	D. $\sim \ln N$
___ *Min* height of left-leaning red-black BST with N keys.	E. $\sim \lg N$
___ *Max* height of left-leaning red-black BST with N keys.	F. $\sim 2 \lg N$
___ *Min* height of a weighted quick union tree with N items.	G. $\sim 2 \ln N$
___ *Max* height of a weighted quick union tree with N items.	H. $\sim N$

ANSWER:

E *Min* height of a binary heap with N keys.	A. ~ 1
E *Max* height of a binary heap with N keys.	B. $\sim \frac{1}{2} \lg N$
C *Min* height of a 2-3 tree with N keys.	C. $\sim \log_3 N$
E *Max* height of a 2-3 tree with N keys.	D. $\sim \ln N$
E *Min* height of left-leaning red-black BST with N keys.	E. $\sim \lg N$
F *Max* height of left-leaning red-black BST with N keys.	F. $\sim 2 \lg N$
A *Min* height of a weighted quick union tree with N items.	G. $\sim 2 \ln N$
E *Max* height of a weighted quick union tree with N items.	H. $\sim N$

6. Suppose that the following keys are inserted into a hash table of size 10 in some order

key	hash
A	5
B	2
C	5
D	1
E	4
F	1
G	3

(a) Give the contents of the linear-probing array if the keys are inserted in alphabetical order: A, B, C, D, E, F, G

	D	**B**	**F**	**E**	**A**	**C**	**G**		
0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**

(b) Which of the following could be the contents of the linear-probing array if the keys are inserted in some other order?

I.

0	1	2	3	4	5	6
A	F	D	B	G	E	C

II.

0	1	2	3	4	5	6
F	A	D	B	G	E	C

III.

0	1	2	3	4	5	6
C	A	B	G	F	E	D

ANSWER
I only.

- I results from inserting the keys in the order B D F A C E G.
- II cannot result. The first key inserted will end up in the table entry corresponding to its hash value. But no key has this property.
- III cannot result. Both A and F end up in the table entry corresponding to their hash values, so we can assume they were inserted first and second. So, the third key inserted will also end up in the table entry corresponding to its hash value. But no keys (beside A and F) have this property.