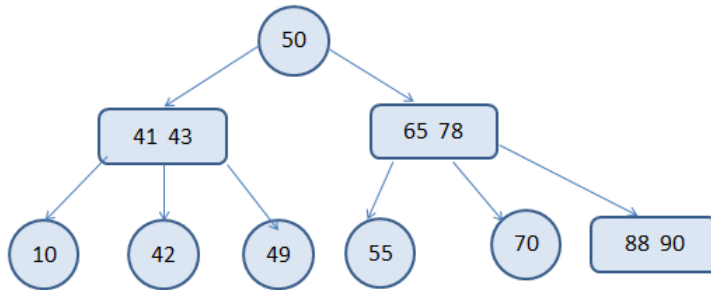


COS 226 Data Structures and Algorithms  
Computer Science Department  
Princeton University  
Fall 2015

## Week 6 Precept

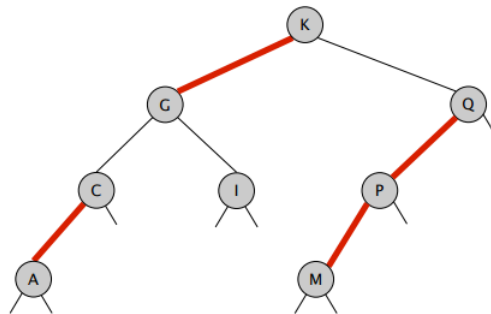
### 1. Left Leaning Red-Black Trees

(a) Consider the following 2-3 Tree.

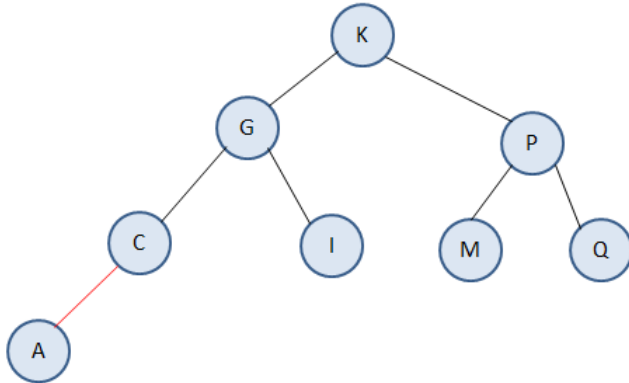


Draw the corresponding red-black tree.

(b) Consider the following red-black tree. Does this tree satisfy LLRB invariants? If not, draw the new tree after fixing the links. Just draw the links that were changed (if any)



- (c) Consider the new red-black tree drawn in Part(b). Draw the resulting red-black tree after inserting key H and then draw it again after inserting key T. The new tree must be a valid red-black tree.



- (d) LLRB construction

Given the keys  $A < B < C < D < E < F < G$ , insert them into a LLRB in the given order showing rotations and flips after each operation. Draw the LLRB after each character insertion.

## 2. Algorithm Design Question

An array  $b$  is called a Circular Shift of array  $a$ , if  $b$  is obtained by rotating a sorted array  $a$  clockwise as shown below.

sorted array a[]										circular shift b[]									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
1	2	3	5	6	8	9	34	55	89	34	55	89	1	2	3	5	6	8	9

- (a) Assume that the array  $b$  consists of  $N$  comparable keys, no two of which are equal. Array  $a$  is not provided. Design an efficient algorithm to determine the minimum value of array  $a$ . Briefly describe your algorithm, using crisp and concise prose.
- (b) Design an efficient algorithm to find any given key in array  $b$ . You can use your algorithm in part (a) to help solve this problem. Briefly describe your algorithm, using crisp and concise prose.

### 3. Midterm Preparation

The following topics will be covered in the midterm exam. Be sure to read lecture notes, review assignments, blackboard exercises. The exam page is available at <http://www.cs.princeton.edu/courses/archive/fall15/cos226/exams/midterm-info-fall.html>. You can find information about the exam location, review sessions and office hours before midterm.

- (a) *Union-Find*: quick-find, quick-union, weighted quick-union
- (b) *Elementary Data structures*: resizing arrays, linked lists, stacks, queues
- (c) *Elementary Sorting Algorithms*: insertion sort, selection sort, Knuth shuffle
- (d) *Linearithmic sorting algorithms*: mergesort, bottom-up mergesort, quicksort, 3-way quicksort, quickselect
- (e) *Priority Queues*: binary heaps, heapsort
- (f) *Binary Search and BST's*: sequential search, binary search, BSTs
- (g) *kd-trees*: kd-trees, interval search trees, 2-3 trees, left-leaning red-black BSTs
- (h) *Hashing (NOT COVERED)*: separate chaining, linear probing