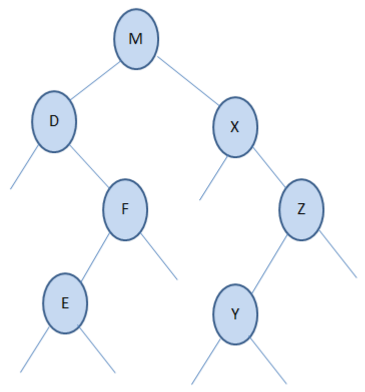
COS 226 Data Structures and Algorithms
Computer Science Department
Princeton University
Fall 2015

Week 5 Precept

1. Binary Search Tree (BST)

   (a) Draw all possible binary search trees (BST's) that can be generated using keys A,B and C. Assume $A < B < C$.

   (b) An inorder traversal of a binary tree is obtained by recursively visiting its left subtree, processing the root and recursively visiting the right subtree. A preorder traversal of a binary tree is obtained by processing the root, and then recursively visiting left and right subtrees. Write down the order of the keys visited in the following BST, during an inorder traversal? A preorder traversal?.



   inorder:________________________

   preorder:________________________

   (c) Consider a BST with the following inorder and preorder traversals.
   inorder: $A\ B\ C\ D\ M\ N\ O$
   preorder: $D\ B\ A\ C\ M\ O\ N$
   Draw a BST that produces these sequences when traversed in the given orders. Is this tree unique?

2. Designing Data Structures
   A generalized heap(Gheap) is a binary heap (max or min) that supports an additional operation of update. The following API is given for Gheap.

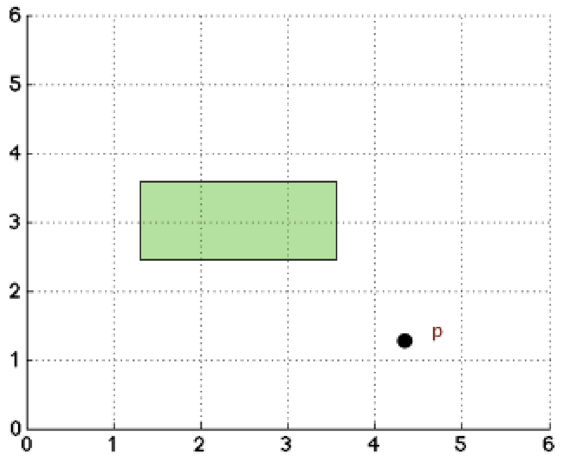
```
public class Gheap<Item item>
{
public Gheap();    /* creates an empty generalized heap */
Item min();       /* returns the minimum item in the heap*/
void insert(Item item);/*insert item to the heap*/
Item update(Item item, int priority);    /*update the priority of item */
Item delMin();   /*delete and return the min item in the heap*/
}
```

   All operations must be supported in logarithmic time or better. Design a data structure(s) to implement a generalized heap. Explain how your data structure will support the required operations in logarithmic time or better. You are allowed to use up to O(N) additional space where N is the heap size.

3. KdTrees

   (a) To the right of the image below, draw the KdTree after inserting the points [(2, 3), (4, 2), (4, 5), (3, 3), (1, 5), (4, 4),(5,4)]. Be careful when inserting (4, 4). Recall that ties are treated the same as greater than.

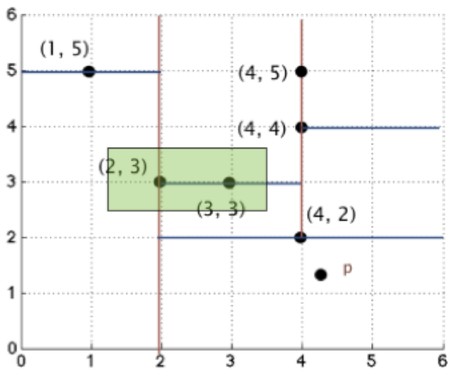
   On the axes, draw each point as well as the red or blue lines that bisect the plane through each point.

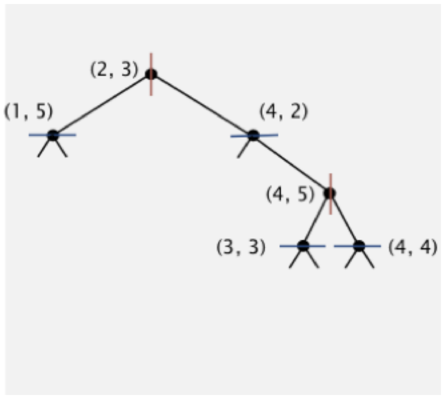


   (b) List the bounding rectangles for each point. The bounding rectangle of (2, 3) is $(-\infty, +\infty)$.

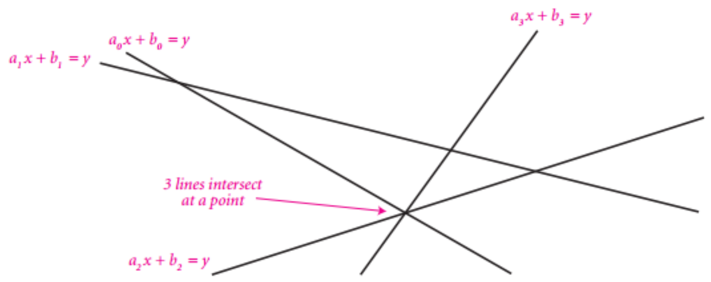
(c) Consider a range query on the shaded rectangle. Write R next to each node in your KdTree (drawn in Part a) that is considered during the KdTree traversal by the range search algorithm. Do not count null nodes or nodes whose corresponding rectangles do not intersect the query rectangle.

(d) Consider a nearest neighbour query on point p. For your convenience, the correct answer from part a (minus the point $(5,4)$) is provided below. First insert the point $(5,4)$ to KdTree and divide the plane and show the bounding box. Then Number each node (starting with 0) by the order in which it is visited by the nearest neighbor algorithm UNLESS that node's corresponding rectangle rules out that node or its children. For nodes that are pruned based on rectangle distance, write an X instead of a number. Do not number null nodes.





(e) KdTree Application (Bonus problem)
Given N lines in the plane, design a $O(N^2 logN)$ algorithm to determine if any 3 (or more) of them intersect at a single point. *For simplicity, assume that all of the lines are distinct and that there are no vertical lines*



$a_1x + b_1 = y$
$a_0x + b_0 = y$
$a_3x + b_3 = y$
3 lines intersect at a point
$a_2x + b_2 = y$