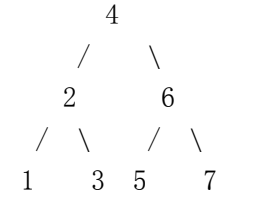
Sample solution

Final Exam, Fall 2002

1.

(a) B_tree of order 4

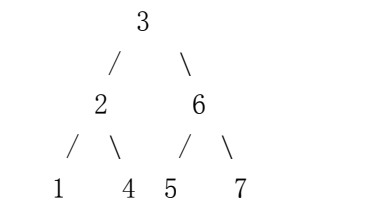
2 <= degree <= 4 1 <= key <= 3



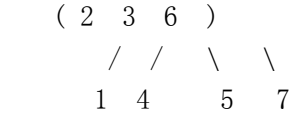
(b)

Using Top-down method.

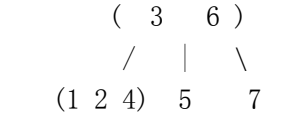
transform to leaf deletion



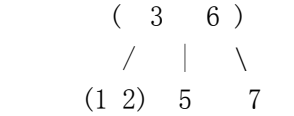
merge nodes 2 3 6



merge element 1 2 4



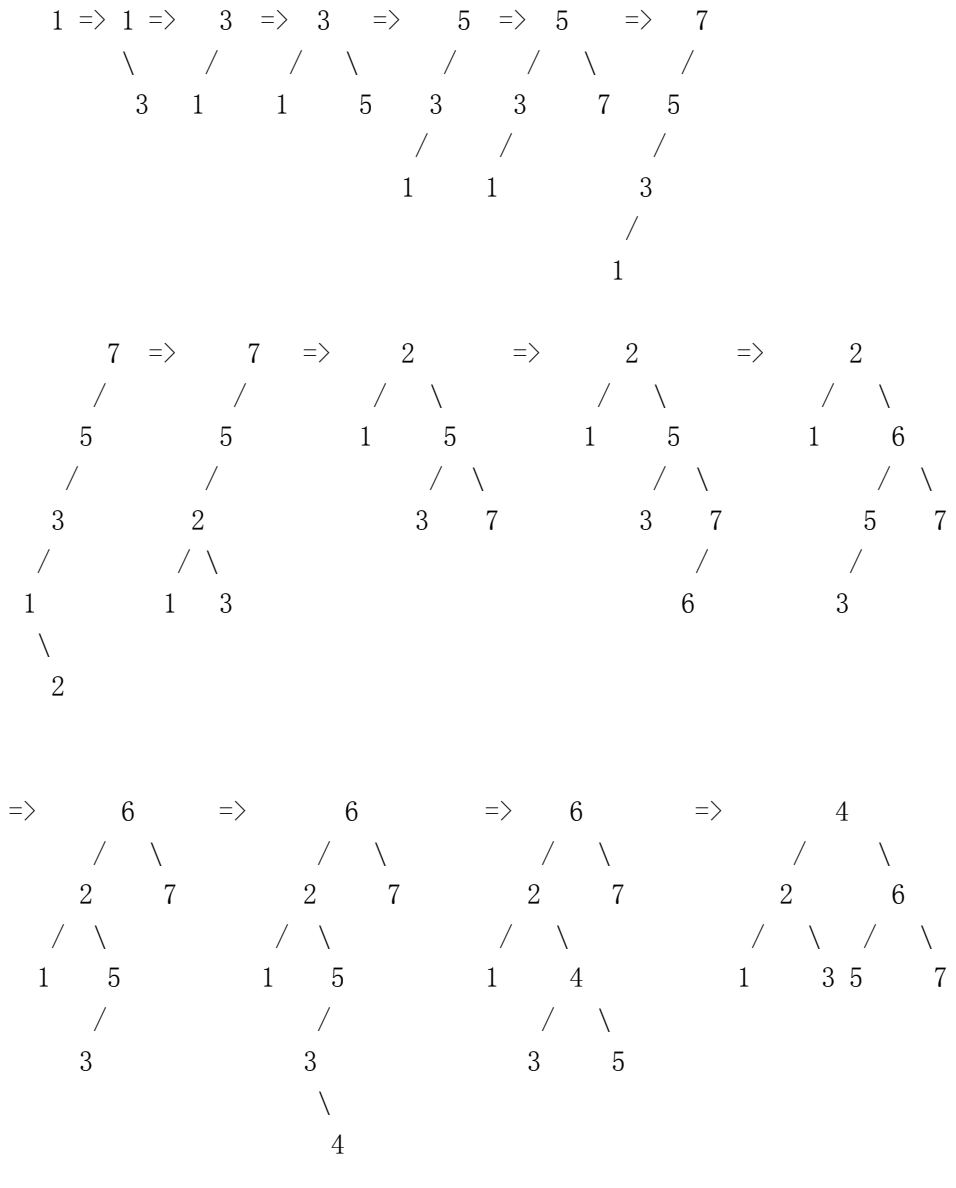
delete 4



2.

Part A:

Answer: 1 3 5 7 2 6 4



Part B:

Splay tree is better when the insertion to the b.s.t results in a highly unbalanced tree.

amortized cost of splay tre : O(log n)

amortized cost of binary search tree : O(n)

Splay is better when the temporial-locality property holds

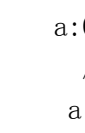
=> splay operation put frequently used element on upper part of tree but binary search tree doesn't.

Splay opreration makes split easy.

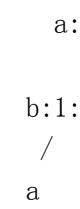
3.

node: <label>:<bitnumber>:<key>.

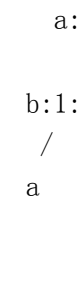
After inserting 0100:



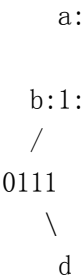
After inserting 1011:



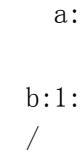
After inserting 1001:

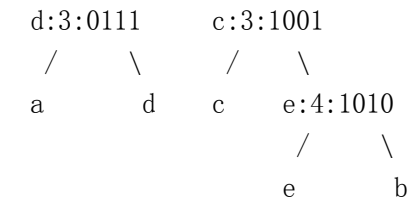


After inserting 0111:

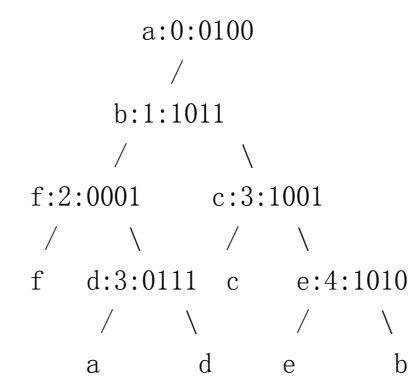


After inserting 1010:



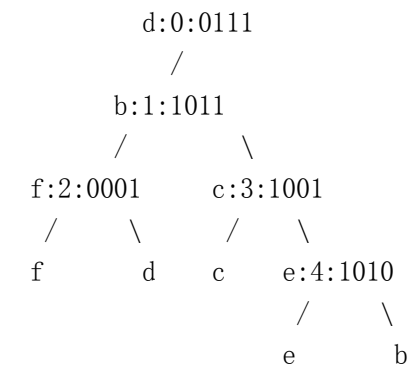


After inserting 0001:



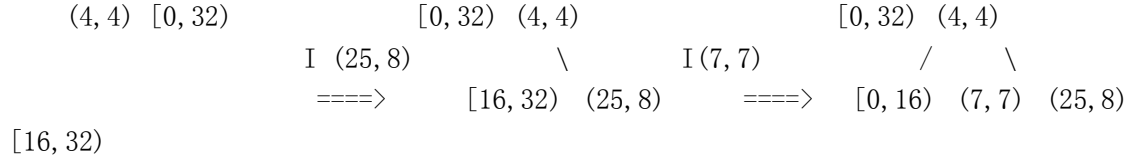
part b:

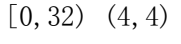
After deleting 0100

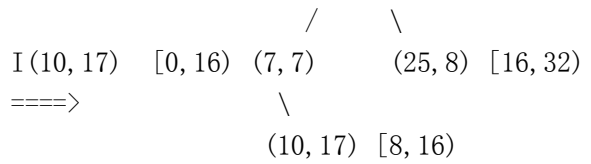


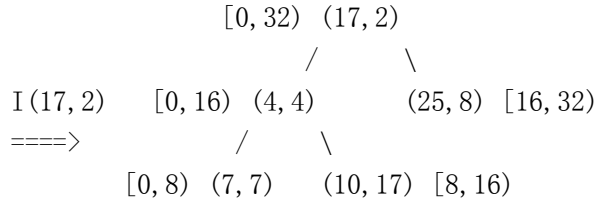
4. Remember that RPST is min tree on y and search tree on x for (x,y)

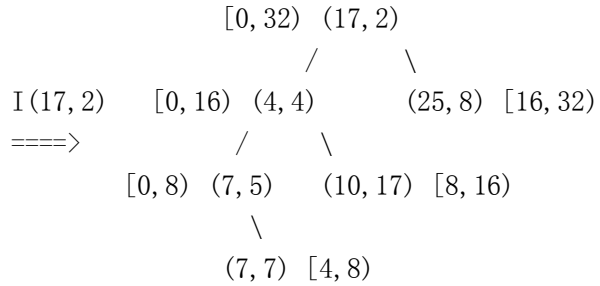
(a)



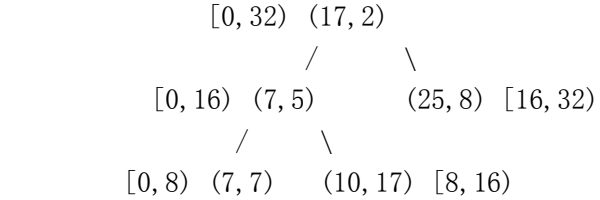








(b) Similart to delete min of min heap



5.

(a) A quad-tree is a degree 4 tree, i.e., each node has up to 4 children and

- Each node represents a portion of an image
- Root node represents entire image
- The children of a node that represents a 2^q * 2^q region represents its 4 2^(q-1) * 2^(q-1) subregions.
- Each node has one of the colors white, black, and gray
- White and black nodes have no children
- Gray nodes have 4 children each

A quad-tree can be used to represents a black and white image.

(b)

Suppose that we have N records with 2 keys, x and y.

A 2-D range tree is a binary tree using the range of x, while in each node we keep a sorted list based on y. The tree always splits the records in half, using a median x key value as the discriminator.

Searching (x,y) can be done as follows:

If x = x-value in the root, search sorted array at the root to search y=y-value.
If x <= x-value in the root, recursively search left subtree
Else if x > x-value in the root, recursively search righ subtree