Projects 1 7 out
2 6
3
Hw 2

Department of Computer Science,
Data Structures (CSC212),
Final Exam
2nd Semester 1429-30H
25/06/1430 (18/06/2009G)

Instructors: Eng. Gamal Shorbagy, Dr. Inayatullah Shah, Dr. Muhammad Hussain
Time: 2 hours **Marks: 100**

Question 1. (4 + (2+2+3+3+6) = 20 Marks)

(a) Draw the graphical representation of a doubly-linked list with sentinel header and trailer nodes, and with the following data elements in it: 4, 5, 2 and 7.

(b) Write the class declaration for the doubly-linked list with sentinel header and trailer nodes, **DoubleLinkListSen**, implementing the ADT List specification, with the following members,

(i) the data members of the class.
(ii) the constructor of the class.
(iii) the method **empty()** that returns true if the list is empty otherwise false.
(iv) the method **last**() that returns true if the current is pointing to the last node otherwise false.
(v) the method **insertAtEnd()** that inserts a new node at the end of the list (as the last node) and makes the current point to the new node.

Question 2. (10 Marks)

Write a client method using the operations of ADT Stack to return the bottom element of a stack.

```
public static <T> T popBottom (Stack<T> S)
```

Preconditions: Stack S is not empty.
Results: The element at the bottom of the stack S is removed and returned. The order of the remaining elements remains unchanged in S.

Question 3. (8+8+4=20 Marks)

(a) You have to store information about each student in a group of about 200 students in a hash table. Each student's key is his id. number, for example, 427102181.

(i) Give a suitable table size and a hash function based on **digit selection**, assuming that the external chaining is the collision resolution strategy employed.
(ii) Give a suitable table size and a hash function based on **division** if the number of students is exactly 200 and linear rehashing is the collision resolution strategy employed.

(b) Insert the following keys: 904, 918, 855, 913, 806, 841 and 778, into a hash table with hash function ***H(key) = key mod 7***, using linear rehashing as collision resolution strategy.

(c) How many probes are required to store 913 and 841?

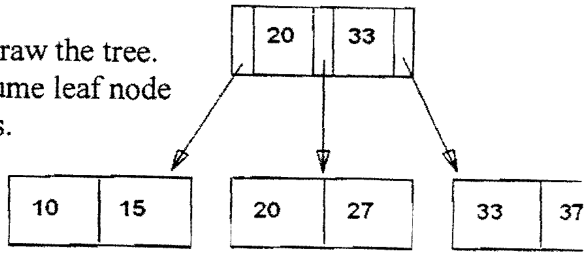
Question 4. (7+8 = 15 Marks)

(a) A binary tree has ten nodes. The inorder and preorder traversal of the tree are shown below. Draw the tree.
Preorder: J C B A D E F I G H
Inorder: A B C E D F J G I H

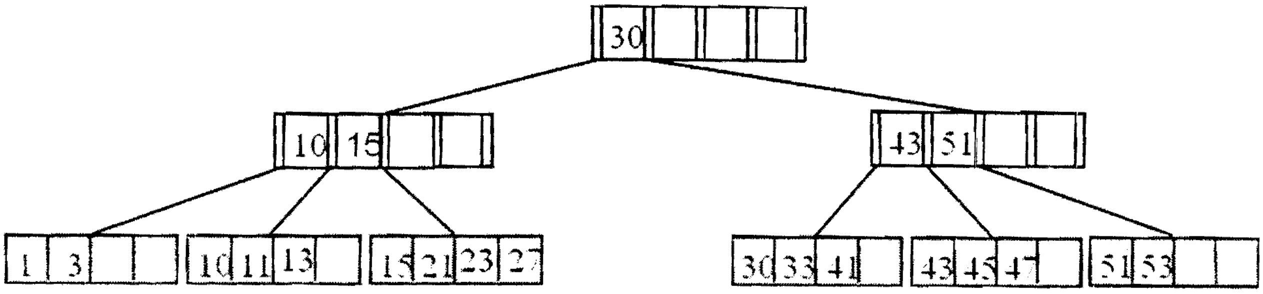
(b) Write a client method that prints keys in a BST in descending order.

Question 5. (8+7 = 15 Marks)

(a) In the following B^+-tree of order 3 insert 24 and redraw the tree. In the new tree, insert 21 and 38 and redraw it. Assume leaf node can have minimum 1 and maximum 2 data elements.

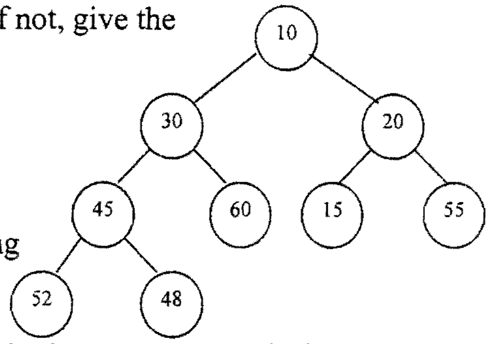


(b) From the B+-tree of order 5 (given below) delete 33, 41 and 11 and redraw the tree.
From the new tree delete 3 and redraw it.
Assume leaf node can have minimum 2 and maximum 4 data elements



Question 6. (3+7+7+3=20 Marks)

(a) Is the binary tree (shown on the right) a min heap? If not, give the reason why not.



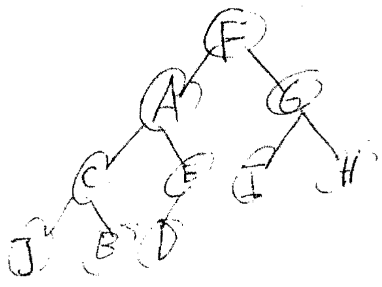
(b) Consider the following array:

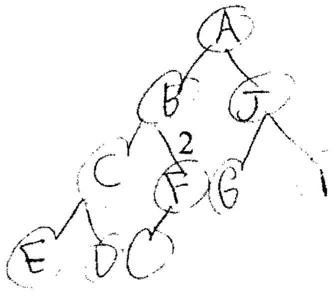
3	15	19	31	47	49	54	57	30	8

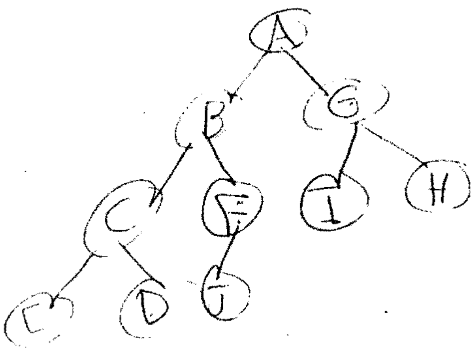
Represent it as a complete binary tree. Is the resulting complete binary tree a max heap? If not using SiftDown operation convert it into a max meap.

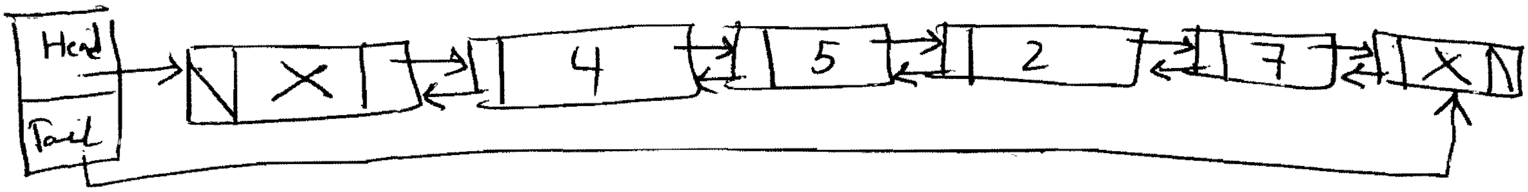
(c) Enqueue the following elements with the given priority into an empty priority queue implemented as a heap: 9, 13, 1, 15, 6, 5, 8, 7, 4, 11, and 20. Assume that a bigger number indicates higher priority e.g. 13 has higher priority than 9. Draw each operation as a binary tree.

(d) In the priority queue obtained in part (b) perform one Dequeue (Serve) operation and show the queue after the operation as a binary tree.









Head
Tail
N
X
4
5
2
7
X
N

Q1

```java
public class DNode <T>
{
  public T data;
  public DNode<T> next,prev;

  public DNode()
  {
        data = null;
        next = prev = null;
  }

  public DNode(T val)
  {
        data = val;
        next = prev = null;
  }
}
```

```java
private DNode<T> head,tail;
private DNode<T> current;

public SentDLinkList()
{
  head = current = new DNode<T>();
  tail = new DNode<T>();
  head.next = tail;
  tail.prev = head;
}

public boolean empty()
{
  return head.next == tail;
}

public boolean last()
{
  return current.next == tail;
}

public void insertAtEnd(T val)
{
  DNode <T> tmp = tail.prev;
  tail.prev = new DNode<T>(val);
  current = tail.prev;
  current.next = tail;
  current.prev = tmp;
  tmp.next = current;
}
```

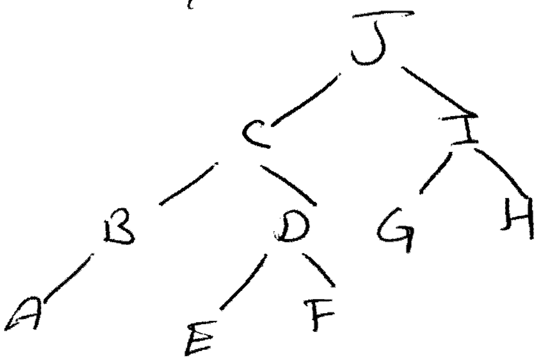
Q2

```
public static <T> T popBottom( Stack <T> S)
{
    Stack <T> S2 = new Stack<T> ();
    T x;
    while ( ! S.empty())
    {   x = S.pop();
        S2.push (x);
    }
    T y = S2.pop();
    while (! S2.empty())
    {   x = S2.pop();
        S.push(x);
    }
    return y;
}
```

Q4
a

preOrder : JCBADEFIGH
InOrder : ABCEDFJGIH

The Binary Tree (BT)



Q4
b

```
public void inOrder(BSTNode<T> P)
{
    if(P != null)
    {
        inOrder(P.right);

        sop(P.data);
        inOrder(P.left);
    }
}
```

Q 5

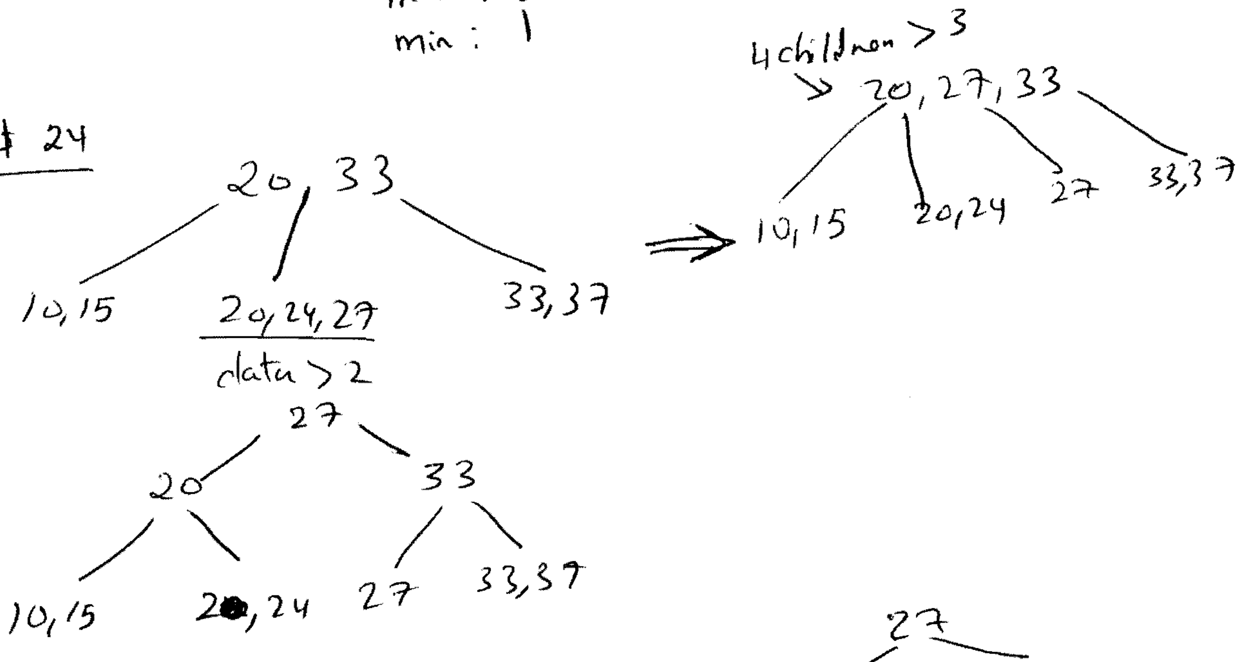
a

Order 3

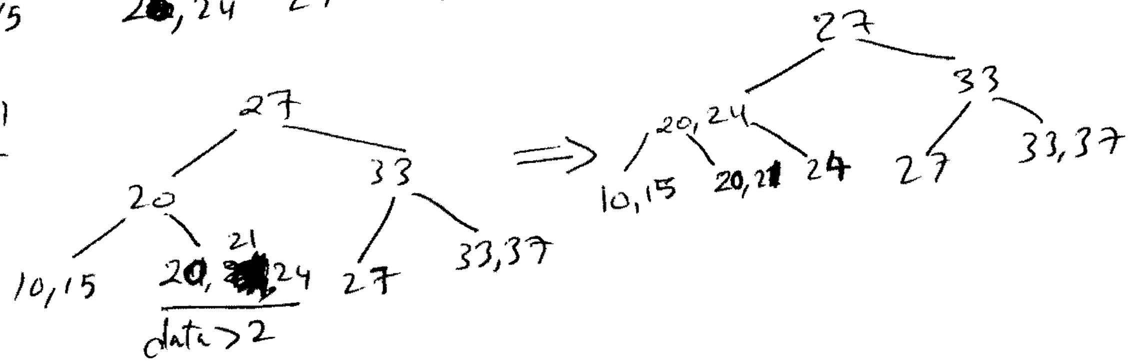
Index node
max : m (3)
min : m/2 (2)

Leaf node
max : 2
min : 1

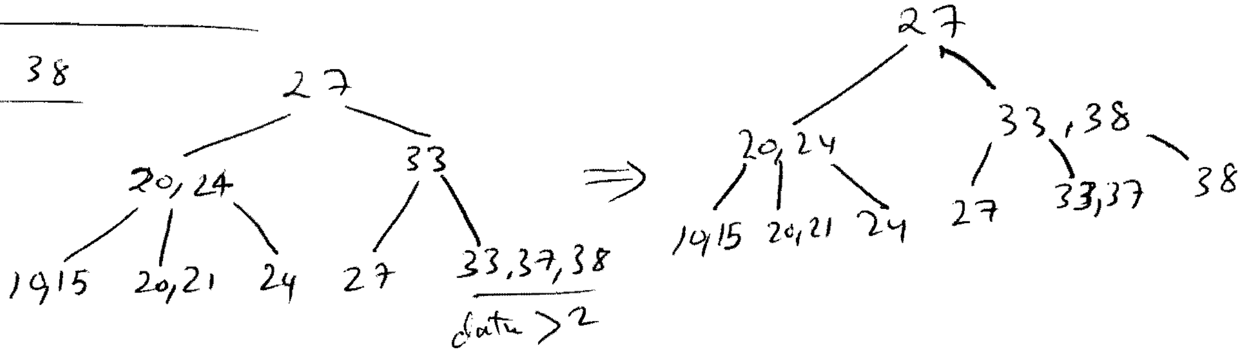
Insert 24



Insert 21



insert 38



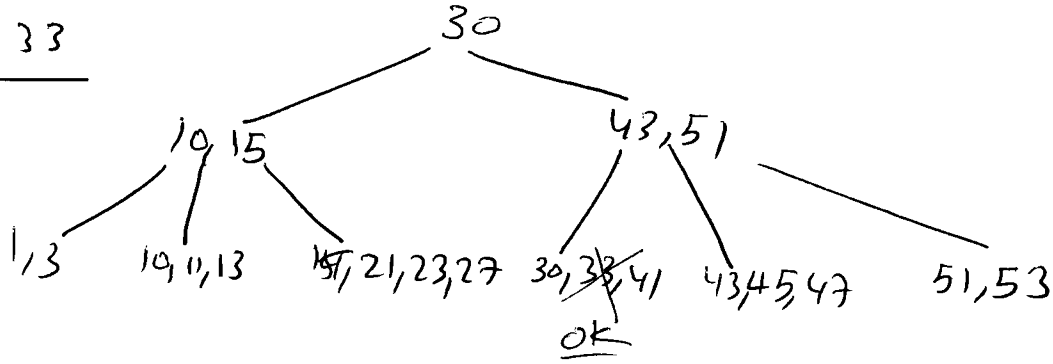
Q5 / 6

Order : 5

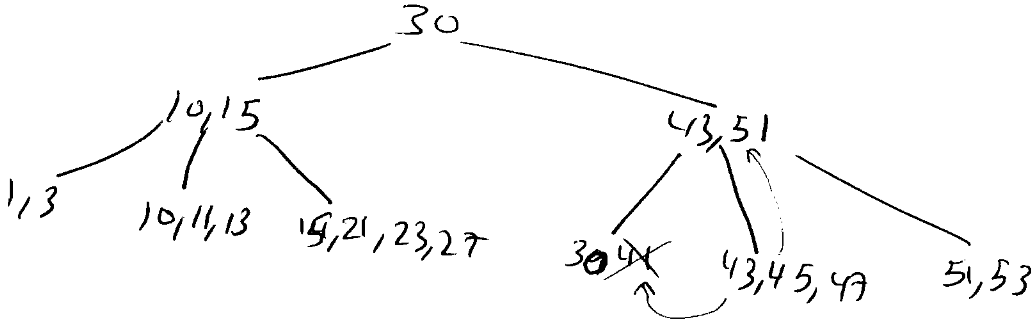
Index : max : 5   min : 3 children

Leaf : max : 4   min : 2 data.

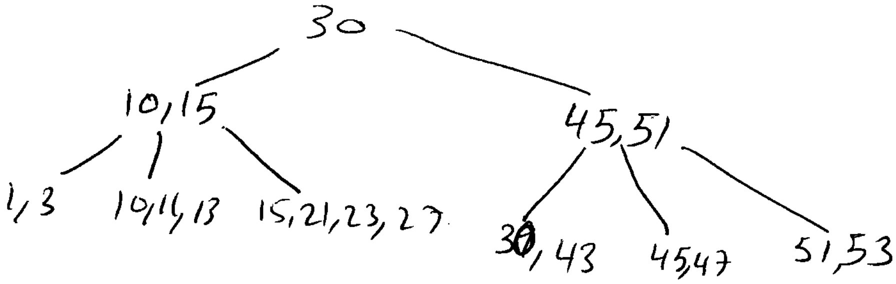
delete 33



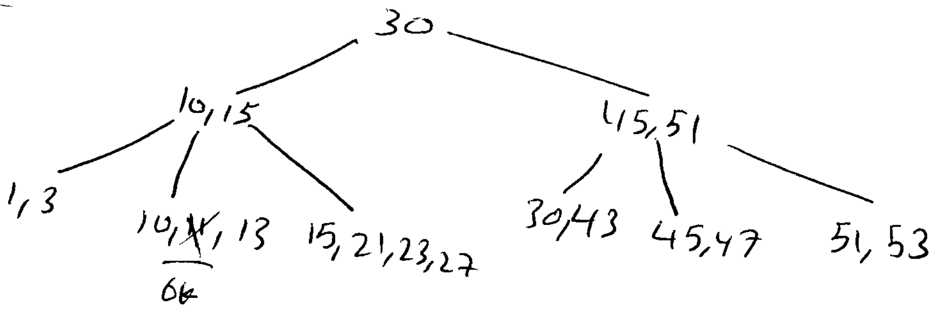
delete 41



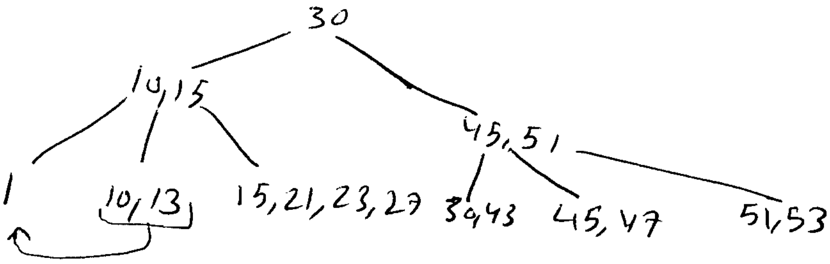
⇒



delete 11

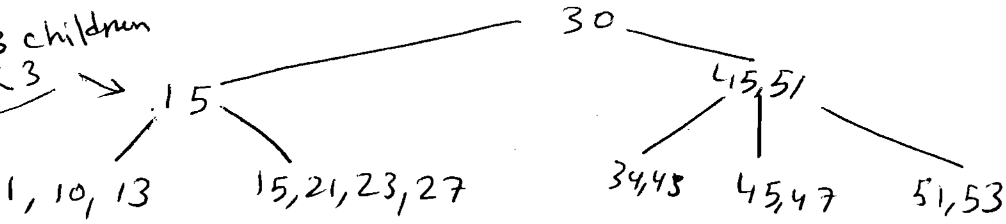


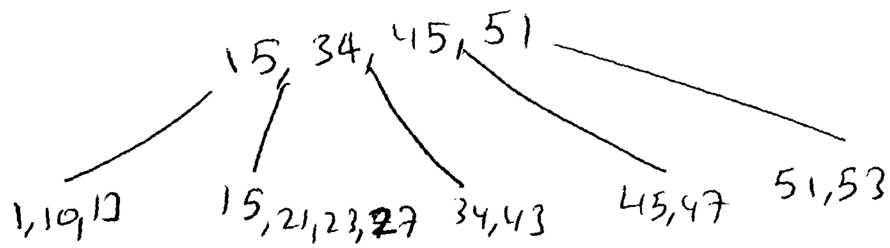
delete 3



min 3 children

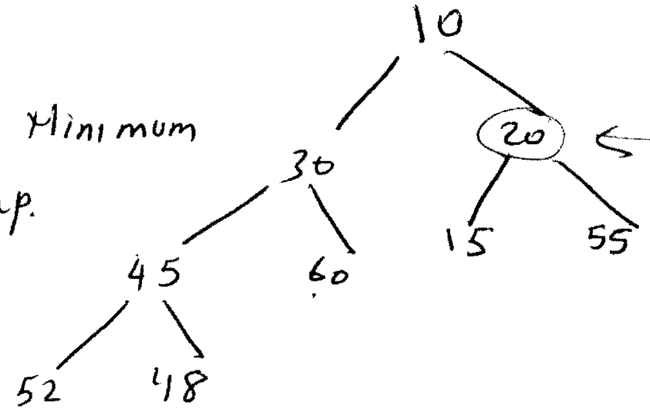
has 2 < 3




15, 34, 45, 51
1, 10, 13
15, 21, 23, 27
34, 43
45, 47
51, 53

Q6
a

NOT Minimum Heap.

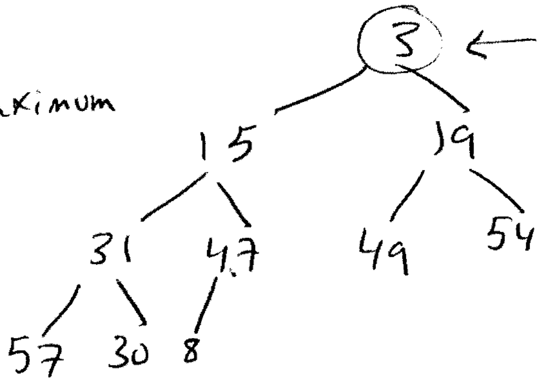


not less then left child

Q6
b

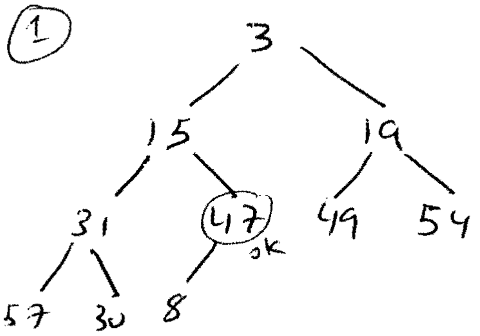
3	15	19	31	47	49	54	57	30	8

NOT Maximum Heap.

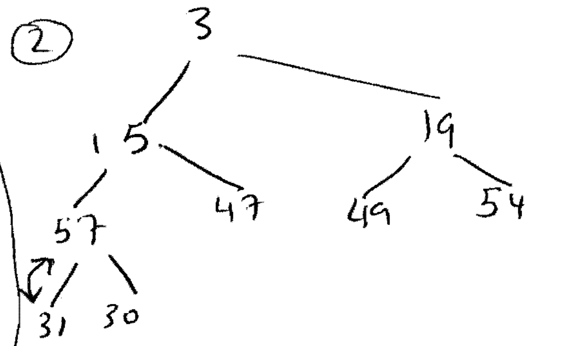


not more then both children.

1

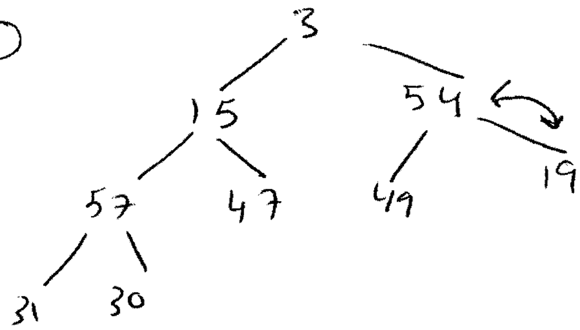


2

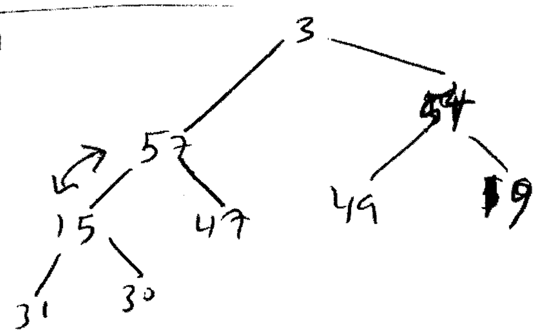


⇒

3

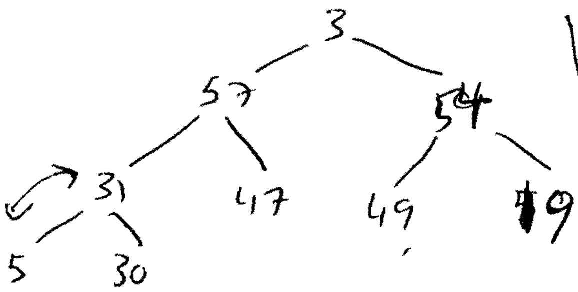


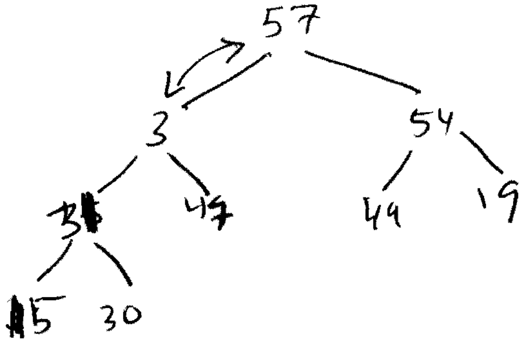
4

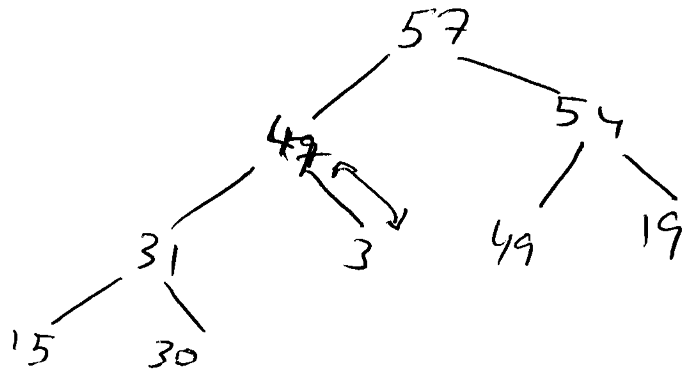


⇒

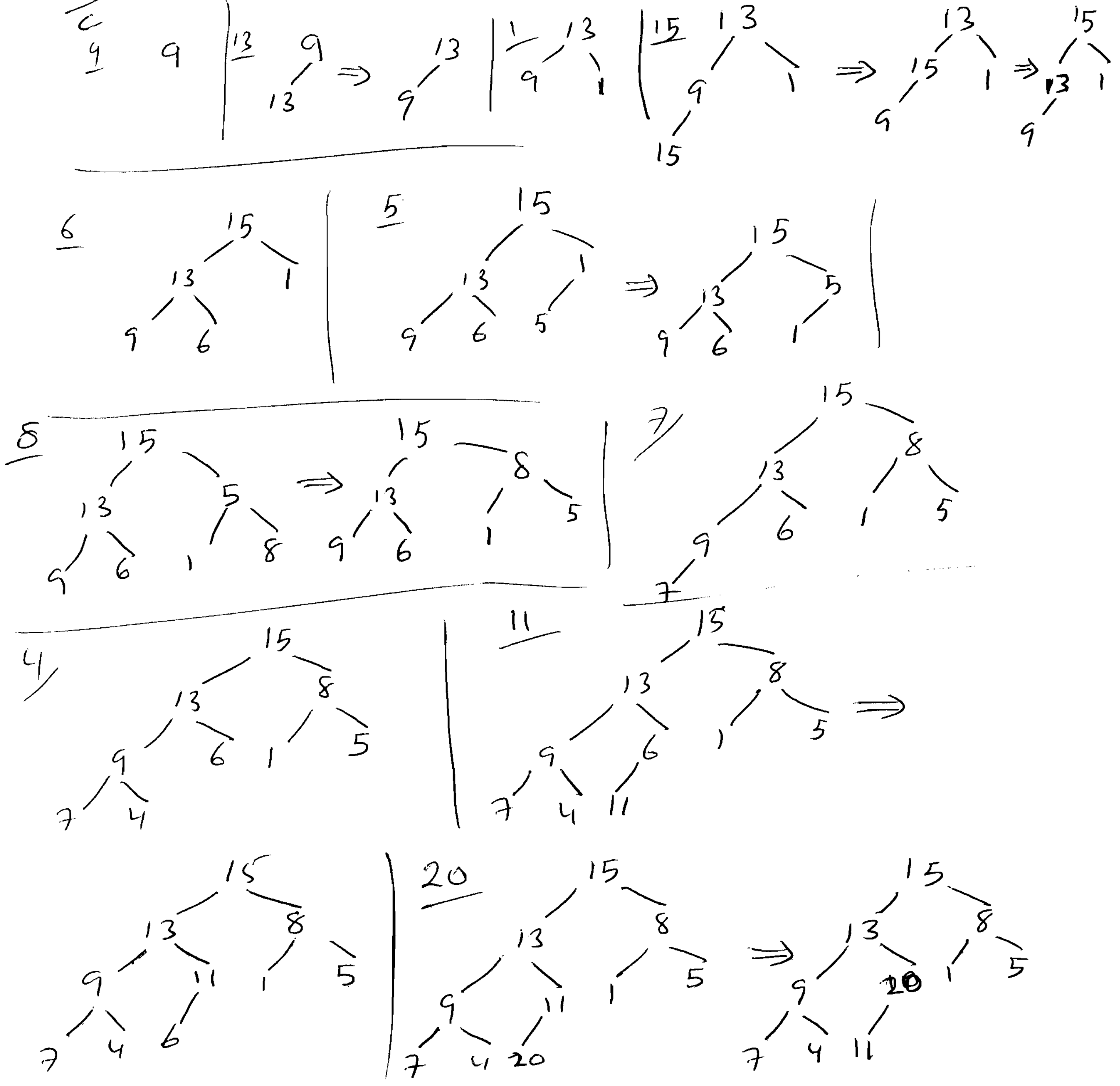
⇒

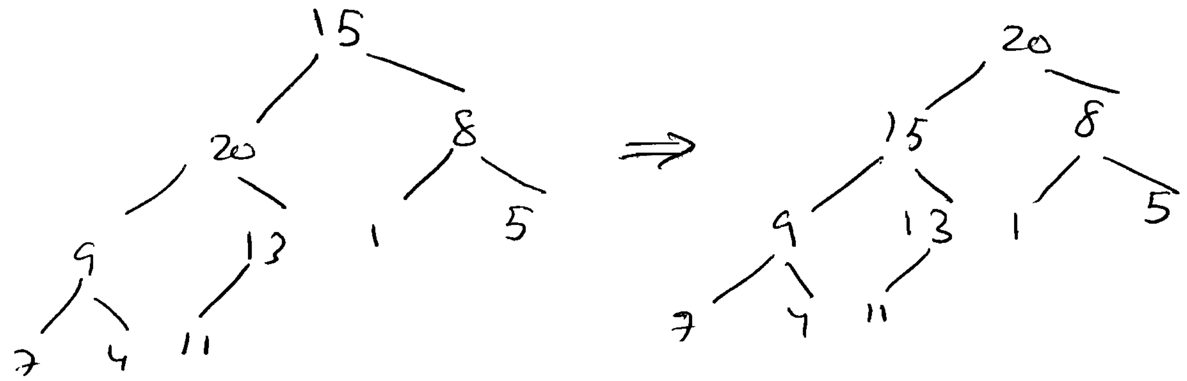






Q 6 c   9, 13, 1, 15, 6, 5, 8, 7, 4, 11 and 20





Q 6/d