

Department of Computer Science
Data Structures (CSC211)
Final Exam (2nd Semester 2006-07)
Date: 21/5/1428H

Time: 3 hours

Marks: 100

Question 1 (15 marks)

Consider the following situations and choose the most suitable ADT that can be used from the choices given.

- (a) You have to store information about books in a library (of 500,000 books). [AVL tree, B+-tree, hashtable, list]
- (b) You have to maintain information about a group of 100 students. No student leaves or enters the group. You often access the information about a student. Each student has unique id number. [AVL-tree, BST, hashtable, stack]
- (c) You have to maintain information about 'father' and 'mother' relations in a large family or a tribe of 200 people. Given a key for a person you should be able to find information about the person's father and mother. Draw a graphical representation of the ADT you choose, labeling it clearly. [BST, AVL-tree, binary tree, B+-tree]

Question 2. (15 marks)

Using the operations in the specification of ADT List write a static method in the main class that inserts a new element in the list as the first element.

```
void insertFirst(List L, Type e)
```

Question 3. (15 marks)

The following method returns the size of a binary tree. Fill in the missing statements: S1, S2 and S3, without declaring any new variables.

```
public int size() {  
    LinkStack<BTNode<T>> s = new LinkStack();  
    BTNode<T> p = root; int size = 0;  
    while (p != null) {  
        S1;  
        S2;  
        S3;  
    }  
    return size;  
}
```

Question 4. (15 marks)

- (a) Enqueue the following elements into a priority queue implemented as a heap and show the resulting priority queue after each insert. Lower value means higher priority. The elements are: {10, 6, 5, 9, 1, 8, 15, 3}.
- (b) From the priority queue obtained in (a) serve an element and show the resulting priority queue.

Please turn over

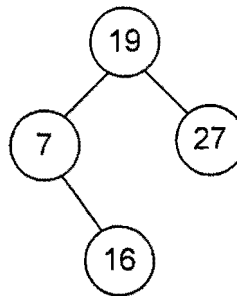
Question 5. (15 marks)

Use the hash function $H(\text{key}) = \text{key} \bmod 11$ to store the sequence of integers: 7, 12, 17, 9, 13, 18, 22, and 25 in a hash table of TableSize = 11. Show the resulting hash table when using:

- (a) Linear rehashing
- (b) External chaining
- (c) Coalesced chaining with cellular size of five and hash function $H(\text{key}) = \text{key} \bmod 6$.

Question 6. (10 marks)

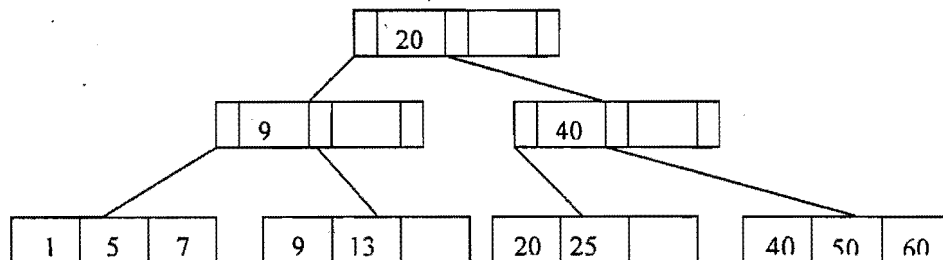
- (a) In the AVL tree shown below insert key 11 and show the resulting tree.
- (b) From the resulting AVL tree of (a) delete key 27 and show the resulting tree.



Question 7. (15 marks)

Shown below is a B+-tree of order 3. Perform the following operations.

- (a) Insert key 70 and show the resulting tree.
- (b) From the resulting tree of (a) delete 13 and show the resulting tree.
- (c) From the resulting tree of (b) delete 9 and show the resulting tree.



Q3/page 5

```
public int size()
{
    LinkStack <BTNode <T>> s = new LinkStack<BTNode <T>>();

    BTNode <T> p = root; int size = 0;

    while(p != null)
    {
        //s1
        if (p.right != null)
            s.push(p.right);

        //s2
        size++;

        //s3
        if (p.left != null)
            p = p.left;
        else if (! s.empty())
            p = s.pop();
        else
            p = null;
    }

    return size;
}
```

Q4-a

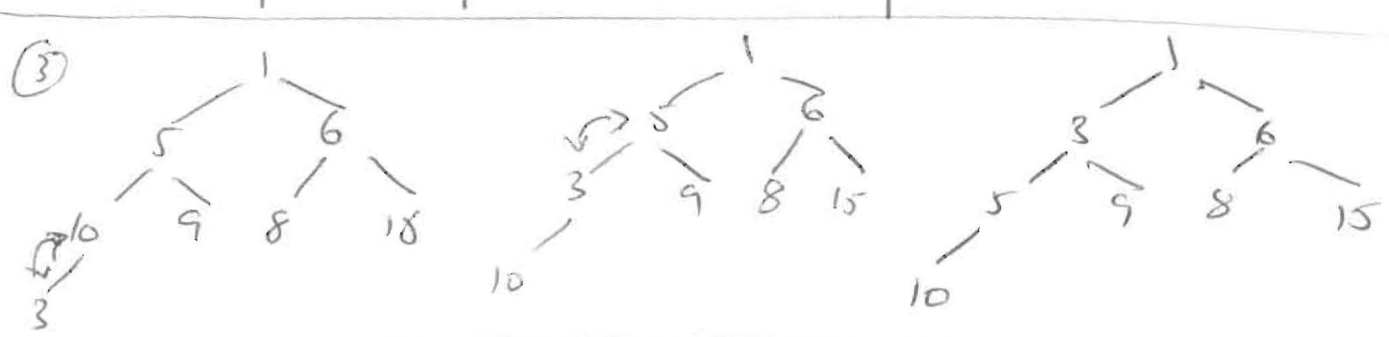
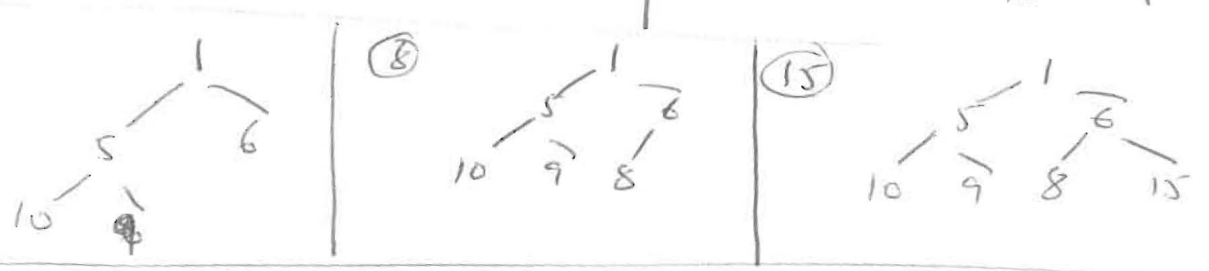
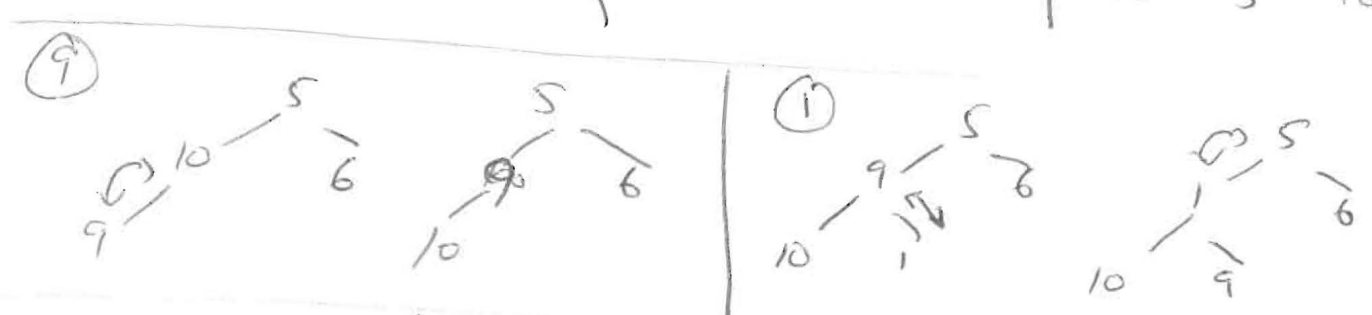
10, 6, 5, 9, 1, 8, 15, 3

(5)

(10)

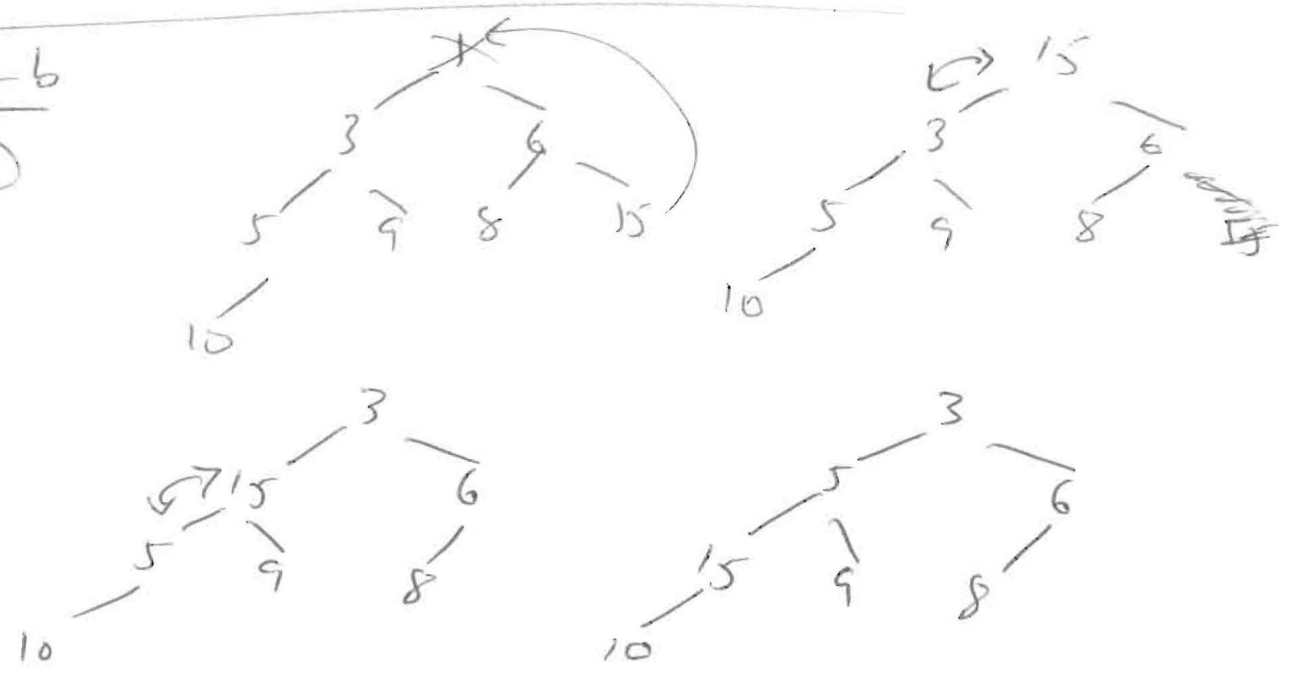
(6)

(5)



Q4-b

(5)

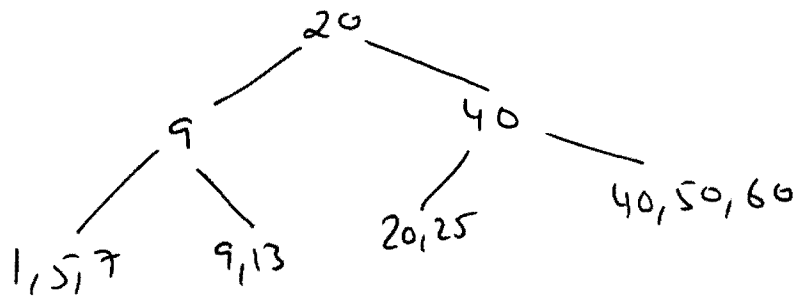


done

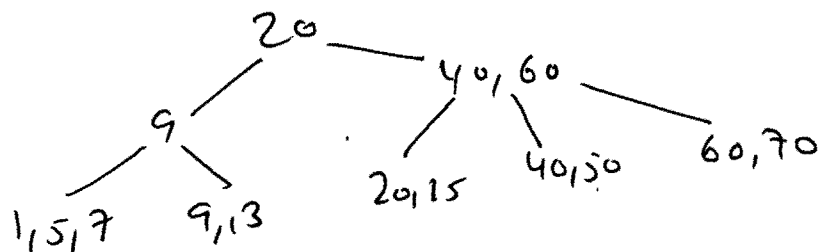
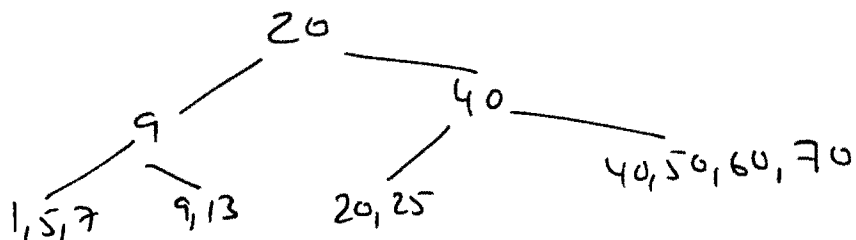
Q7 - a
page 6

Original tree

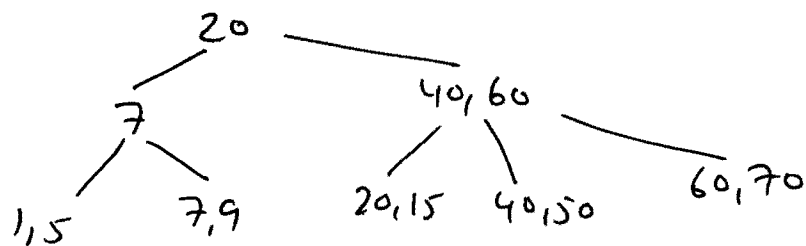
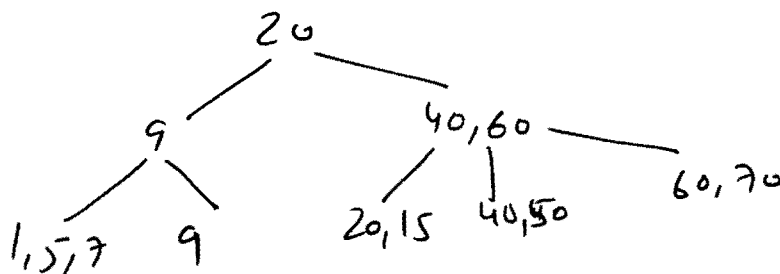
order = 3



Insert 70



delete 13



delete 9

