CSC 212 Final Solution - Fall 2014

College of Computer and Information Sciences, King Saud University
Exam Duration: 3 Hours

03/01/2015

Question 1 [16 points]

1. **(a)** $3n^2 + 10n\log(n) \in O(n^2)$, **(b)** $100n + n^2 + n^3 \in O(n^3)$, **(c)** $2^n + n^2 + n^2\log(n^2) \in 2^n$, **(d)** $n^2 + n\log(n^n) + 2^{\log n} \in O(n^2 \log n)$. The functions in increasing order of growth rate: (a), (d), (b), (c).
2.

```
public void removeBetween(T e1, T e2) {
        Node<T> p = head;
        while ((p != null) && (!p.data.equals(e1)))
                p = p.next;
        if (p == null)
                return;
        Node<T> q = p.next;
        while ((q != null) && (!q.data.equals(e2)))
                q = q.next;
        if (q == null)
                return;
        p.next = q;
        q.previous = p;
        current = head;
}
```

.....

Question 2 [16 points]

1.

```
public void removeHP() {
        if (head == null)
                return;
        int pr = head.pr;
        while ((head != null) && (head.pr == pr)) {
                head = head.next;
                size--;
        }
}
```

2.

```
public <T> boolean isReverse(Stack<T> st1, Stack<T> st2) {
        Stack<T> st3 = new Stack<T>();
        Stack<T> st4 = new Stack<T>();
        while (!st2.empty()) {
                st3.push(st2.pop());
        }
        boolean equal = true;
```

```
        while (!st1.empty() && !st3.empty()) {
                        T e1 = st1.pop();
                        st4.push(e1);
                        T e2 = st3.pop();
                        st2.push(e2);
                        if (!e1.equals(e2)) {
                                        equal = false;
                                        break;
                        }
        }
        if (!st1.empty() || !st3.empty())
                equal = false;
        while (!st3.empty()) {
                st2.push(st3.pop());
        }
        while (!st4.empty()) {
                st1.push(st4.pop());
        }
        return equal;
}
```

.....

Question 3 [16 points]

1.
```
public int twoChildren() {
        return recTwoChildren(root);
}
private int recTwoChildren(BSTNode<T> p) {
        if (p == null)
                return 0;
        if (p.left != null && p.right != null)
                return 1 + recTwoChildren(p.left) + recTwoChildren(p.right);
        return recTwoChildren(p.left) + recTwoChildren(p.right);
}
```

2.
```
public boolean isLeaf(int key) {
        return recIsLeaf(key, root);
}
private boolean recIsLeaf(int key, BSTNode<T> p) {
        if (p == null)
                return false;
        if (p.key == key){
                if (p.left == null && p.right == null)
                        return true;
                else
                        return false;
        }
        if (p.key > key)
                return recIsLeaf(key, p.left);
        else
                return recIsLeaf(key, p.right);
}
```

.....

Question 4 [12 points]

1. Answer:

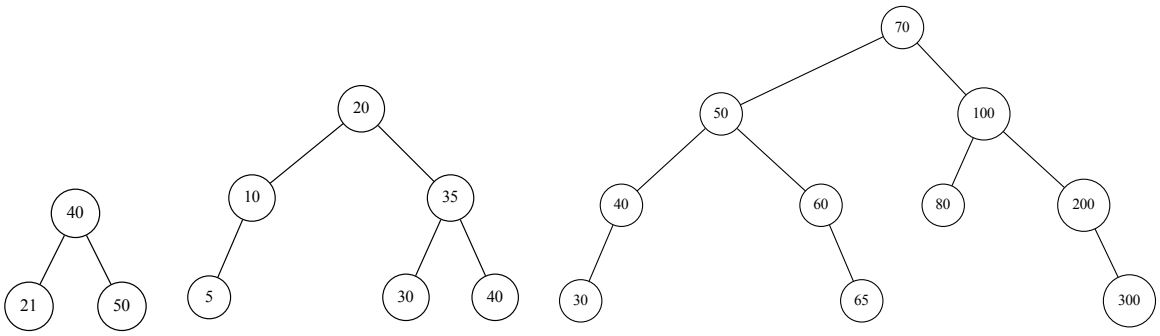
1. Heap after inserting 6: (d) 5,10,6,16,12,8
2. Heap after inserting 10: (c) 5,10,8,16,12,10
3. Heap after inserting 3: (e) 3,10,5,16,12,8
4. Heap after inserting 5: (c) 5,10,5,16,12,8
5. Heap after inserting 8: (a) 5,10,8,16,12,8

2. Answer:

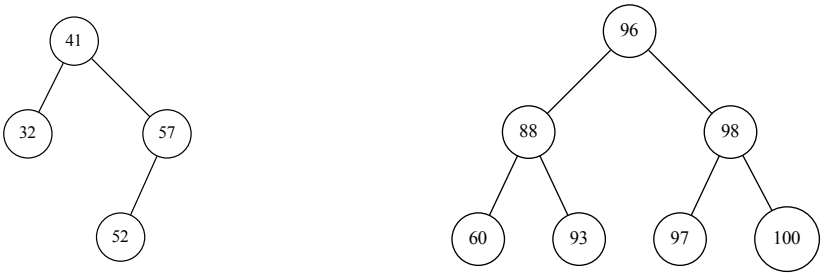
1. After deleting one key: (c) 12,8,4,4,6,2
2. After deleting two keys: (e) 8,6,4,4,2
3. After deleting three keys: (d) 6,4,4,2

. . . . . .

Question 5 [12 points]



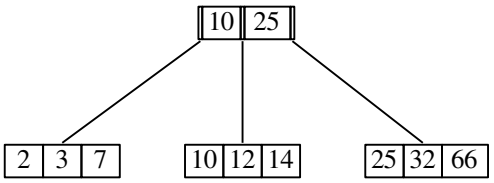
Insert 40 Insert 5 Insert 65



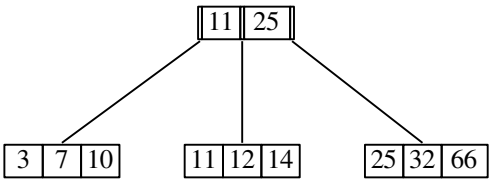
Delete 43 Delete 74

. . . . . .

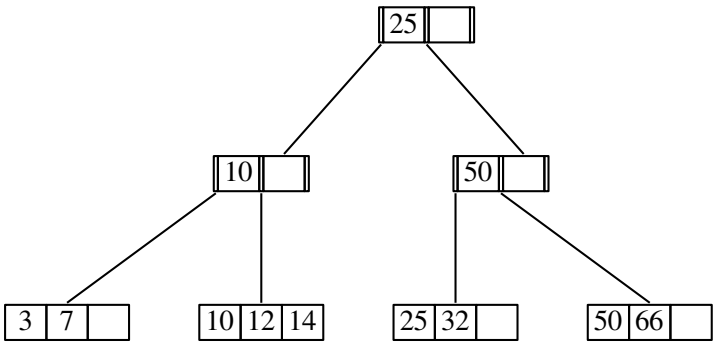
Question 6 [10 points]



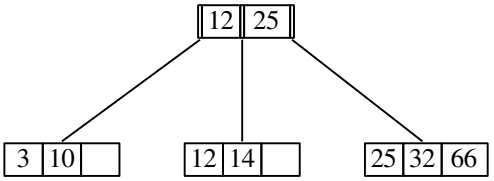
Insert 2



Insert 11



Insert 50



Delete 7

. . . . . .

Question 7 [12 points]

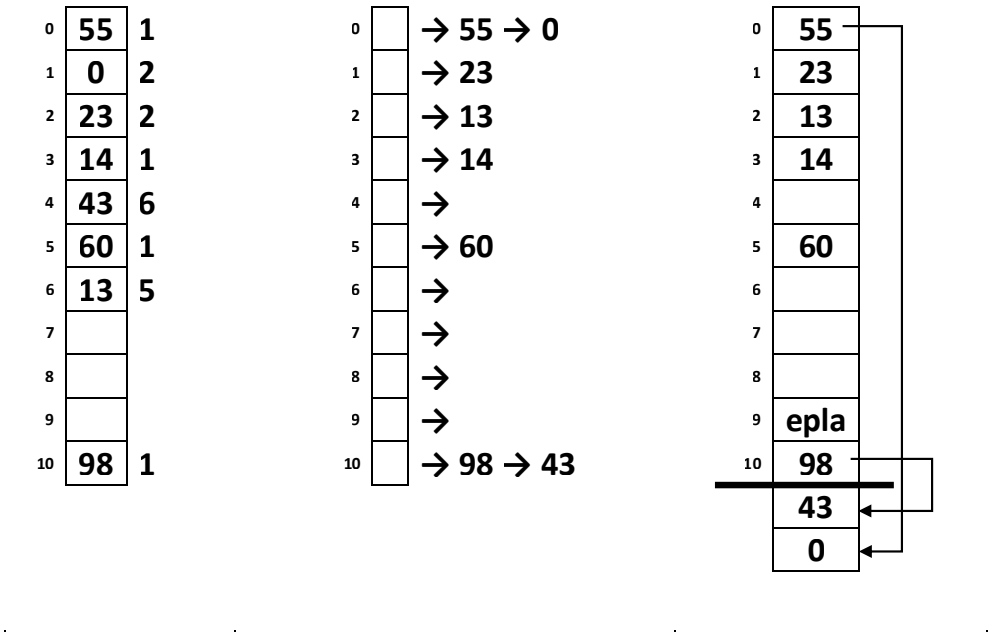


Figure 1: From left to right: linear rehashing, external chaining and coalesced chaining.

. . . . . .

Question 8 [6 points]

0	1	0	1	0
1	0	1	0	1
0	1	0	0	1
1	0	0	0	1
0	1	1	1	0

. . . . . .