Name __Key__   Andrew ID _____

Total points = 131. Score will be a percentage of 131.

Tracing Lists and Trees (28 points)

1.   You will be asked to show the exact output of the following program. (18 Points)

```java
package dsmidterm;

class Node {
    private int data;
    private Node next;

    public int getData() {
        return data;
    }

    public Node getNext() {
        return next;
    }

    public void setData(int data) {
        this.data = data;
    }

    public void setNext(Node next) {
        this.next = next;
    }

    public Node(int data, Node next) {
        this.data = data;
        this.next = next;
    }
}
class List {
    Node head;
    Node tail;
    public List() {
        head = null;
        tail = null;
    }
    public void add(int x) {
        if(head == null) {
            head = new Node(x,head);
            tail = head;
        }
        else {
            tail.setNext(new Node(x,null));
            tail = tail.getNext();
        }
    }

    public boolean isEmpty () {
        return head == null;
    }
```

```java
// (1.a) Pre:
public int remove() {
    int x = head.getData();
    head = head.getNext();
    return x;
}

// (1.b) Big Theta:
public String toString() {
    Node v = head;
    String s = "";
    while(v != null) {
        s = s + v.getData() + " ";
        v = v.getNext();
    }
    return s;
}
// (1.c) Big Theta:
public void foo() {
    tail = head;
    Node currNode = head;
    Node nextNode = null;
    Node prevNode = null;
    while(currNode != null) {
        nextNode = currNode.getNext();
        currNode.setNext(prevNode);
        prevNode = currNode;
        currNode = nextNode;
    }
    head = prevNode;
}

}

public class DSMidterm {

    public static void main(String[] args) {
        List a = new List();
        a.add(9);
        a.add(8);
        a.add(7);
        a.add(6);
        System.out.println(a);    // (1.d) Output of println
        while(!a.isEmpty()) {
            System.out.print(a.remove() + " ");
        }                    // (1.e) Output of all the while() print
        a.add(9);
        a.add(8);
        a.add(7);
        a.add(6);
        a.foo();
        a.add(100);
        System.out.println(a); // (1.f) Output of println
    }
}
```

*head != null or
list is not empty*

1.a Give the proper pre-condition at the point of code marked (1.a). _head != null or list is not empty_ (2 pts)
1.b Give the Big Theta of the code marked (1.b) (toString). $\Theta(N)$ (1 pts)
1.c Give the Big Theta of the code marked (1.c) (toString). $\Theta(N)$ (1 pts)
1.d Show the output of the code marked (1.d) _9 3 7 6_ (4 Points)
1.e Show the output of the code marked (1.e) _9 8 7 6_ (4 Points)
1.f Show the output of the code marked (1.f) _6 7 8 9 100_ (4 Points)

1.g Is it correct to say that the method foo() of the list class runs in $\Omega(2^n)$? Circle True of (False) (1 pt.)

1.h Is it correct to say that the method foo() of the list class runs in $\Omega(1)$? Circle (True) or False (1 pt.)

2.   Study the execution of the following program. Questions appear below. (10 points):

```java
class Node {
    public int data;
    public Node lc;
    public Node rc;
    public Node p;
    public Node(Node lc, int x, Node rc, Node p) {
        this.lc = lc;
        this.data = x;
        this.rc = rc;
        this.p = p;
    }
}

public class SimpleTree {

    public Node root;
    public Node lowNode;

    public SimpleTree() {
        root = null;
        lowNode = null;
    }

    public void add(int x){

        if (root == null) {
            root = new Node(null,x,null,null);
            lowNode = root;
        }
        else {
            Node t = root;
            Node q = t;
            while(t != null) {
                if(x < t.data ) {
                    q = t;
                    t = t.lc;
                }
                else {
                    q = t;
                    t = t.rc;
                }
```

```
        } // end while
        if(x < q.data) {
            q.lc = new Node(null, x, null, q);
            lowNode = q.lc;
        }
        else {
            q.rc = new Node(null, x, null, q);
            lowNode = q.rc;
        }
    }
}
public void traversal(Node r) {
    if(r == null) return;
    if(r.lc != null)traversal(r.lc);
    System.out.println(r.data);
    if(r.rc != null)traversal(r.rc);
}
public void traversal() {
    traversal(root);
}

public void traversal2() {
    Node q = lowNode;
    while( q != null) {
        System.out.println(q.data);
        q = q.p;
    }
}
public static void main(String[] args) {
    SimpleTree st = new SimpleTree();
    st.add(10);
    st.add(20);
    st.add(30);
    st.add(40);
    st.add(2);
    st.add(5);
    st.add(8);
    st.add(9);
    System.out.println("(2.a)");
    st.traversal();
    System.out.println("(2.b");
    st.traversal2();
    st.add(1);
    System.out.println("(2.c)");
    st.traversal2();

}
}
```

2.a What will the program display at the traversal marked Question 2.a? (2 Points)

2 5 8 9 10 20 30 40

2.b What will the program display at the traversal marked Question 2.b? (4 Points)
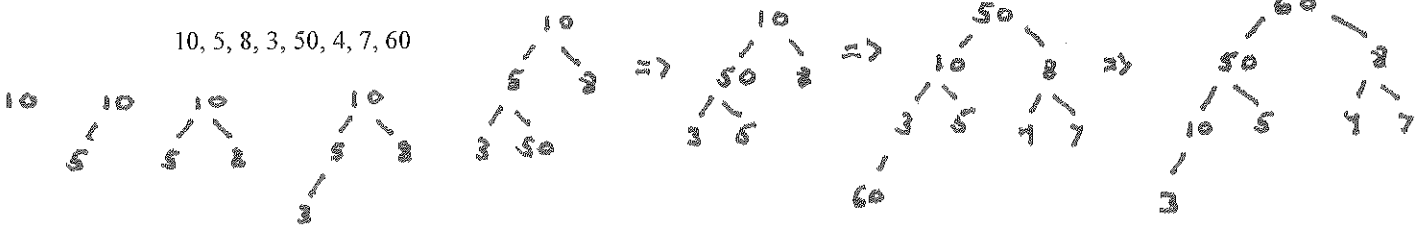
9 8 5 2 10

2.c What will the program display at the traversal marked Question 2.c? (4 Points)

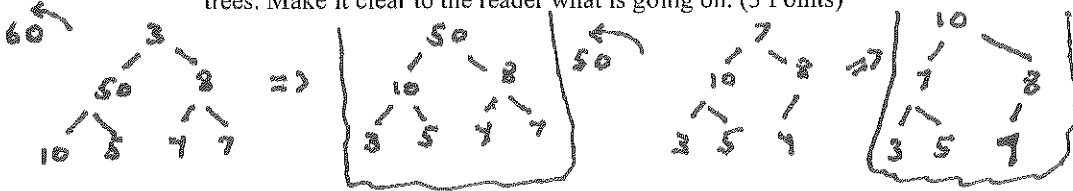1 2 10

Heaps (12 points)

3) Insert the following 8 numbers into a <u>max heap</u>. Draw a new tree for each heap insertion. (4 Points)

10, 5, 8, 3, 50, 4, 7, 60

4) What is the height of the tree that you drew in question 3? (2 Points) ___3___

5) Perform exactly two deleteMax() operations on the heap that you drew in question 3. Draw the resulting trees. Make it clear to the reader what is going on. (3 Points)


6) Consider the following <u>min</u> heap implemented in an array. It is not quite correct. To make it a proper min heap exactly one swap must occur. <u>What two numbers</u> (child and parent) need to be swapped in order to make this a min heap? (3 points). PLACE CHECK MARKS NEXT TO THE TWO NUMBERS THAT NEED TO BE SWAPPED.

2
99 ✓
14
8 ✓
9
15
87
100
102
41
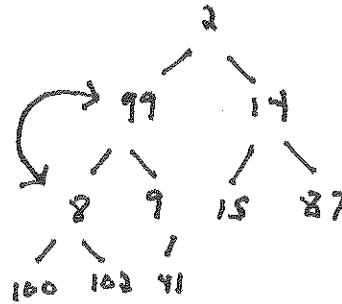
Binary Trees (16 points)

7. Parts (a), (b), (c) refer to the following binary tree:

```
              20
             /    \
           40      97
           /         \
          6           76
           \          /
            8        3
```
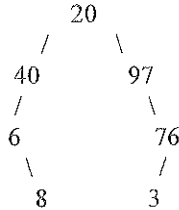
(a) List the data that would be accessed by a pre-order traversal on the given tree by writing out the values in the nodes as they would be accessed, separated by commas. (3 points)

20, 40, 6, 8, 97, 76, 3

(b) List the data that would be accessed by an in-order traversal on the given tree by writing out the values in the nodes as they would be accessed, separated by commas. (2 points)

6, 8, 40, 20, 97, 3, 76

(c) List the data that would be accessed by a level-order traversal on the given tree by writing out the values in the nodes as they would be accessed, separated by commas. (2 points)

20, 40, 97, 6, 76, 8, 3

(d) In general, if a binary (at most two children per node) tree is perfectly balanced (unlike the tree pictured here) and complete with height h, how many nodes, in terms of h, will the tree have? (2 points) $2^{h+1} - 1$ Note, this tree has a perfectly flat bottom. We need the total number of nodes in terms of h.

(e) In general, if a binary tree is perfectly balanced (unlike the tree pictured here) and complete with exactly k leaves. What is the height (in terms of k) of this tree? (2 points) $\log_2 k$ or $\lg k$ Note, this tree has a perfectly flat bottom. MISSING base -1

8. (a) Insert he following numbers into a Binary Search Tree. Draw the tree after all insertions are complete. (3 Points)
10, 20, 30, 40, 50, 60, 5, 35, 25, 32, 28
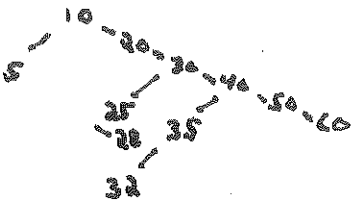
(b) Delete 30 from the final tree that you drew in 8 (a). Draw this final tree. (2 Points)
There are two possible answers. Either one is fine.

Key 7
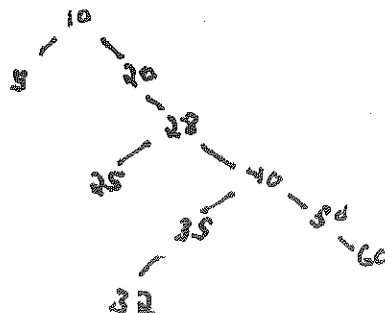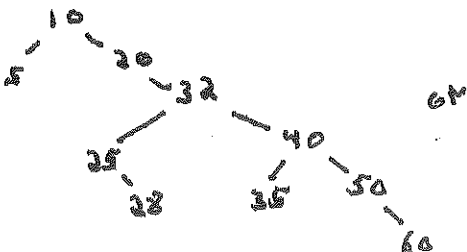
Project Questions (20 points)

(9) Recall the Merkle-Hellman cryptosystem that we worked with in Project 1.

Project 1 was based on the subset sum problem which is known to be NP-Complete. The problem itself can be described as follows: given a set of numbers X and a number k, is there a subset of X, which sums to k?

(a) Suppose X = {56,2,9,12,4,7,234, 5, 1} and k = 23. Is there a subset of X which sums to k?
   yes   (Yes)/No (1 point)

(b) The type of problem you were asked to solve in question 9 (a) is (Circle one answer): (1 Point)

   1. an optimization problem.
   2. a problem that is impossible to solve.
   3. a problem that has been proven to take exponential time to solve.
   4. a problem that has been proven to take factorial time to solve.
   (5.) a decision problem.

(c) Suppose Alice sends a message (K) to Bob. K is computed using Bob's Merkle-Hellman public key combined with the message M. The central idea behind Merkle-Hellman is that a potential eaves dropper could read the message M if the eaves dropper could (circle the one best option) (2 Points)

   1. Find K so that M is prime.
   2. Modify Bob's public key.
   3. Modify the super increasing sequence.
   4. Find a subset of a super increasing sequence that sums to K.
   (5.) Find a subset of Bob's public key that sums to K.

(d) Recall that a modular inverse of an integer b mod m is the integer b^{-1} such that $(b * b^{-1})$ mod m=1. What is the modular inverse of 2 mod 7? __4__ (1 Points)

(e) The following points, in a standard (x,y) coordinate plane, have been added to a 2-d tree. (3,4), (5,2), (4,1). The 2-d tree appears as follows:

```
(3,4)
   \
   (5,2)
   /
(4,1)
```
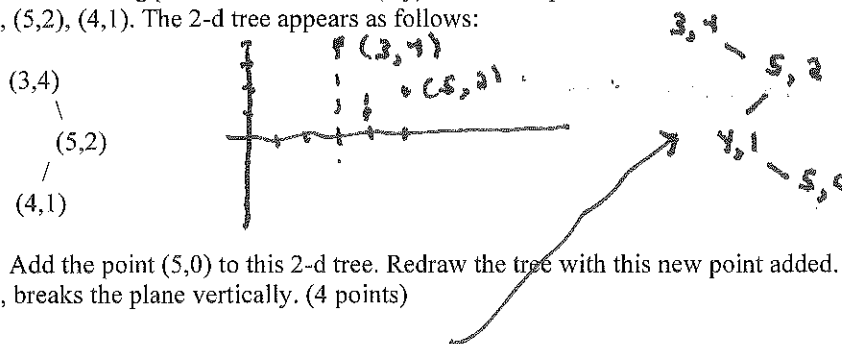

   Add the point (5,0) to this 2-d tree. Redraw the tree with this new point added. The first point, (3,4), breaks the plane vertically. (4 points)

(f) In Project 3 we wrote a Red Black binary search tree. Suppose we are doing an insert of a variable name into a Red Black Tree. Let T(n) be the number of operations required to do the insert. In the worst case, which of the following are true about T(n)? Circle all of those that are true. (You may or may not have more than one answer.) (4 Points)

-1 for each error
max off is 4.

   (1.) $T(n) \varepsilon \, O(Log(n))$
   2. $T(n) \varepsilon \, O(1)$
   3. $T(n) \varepsilon \, \Omega(n^2)$
   (4.) $T(n) \varepsilon \, O(n)$

⑤ $T(n) \; \varepsilon \; \Theta(Logn)$
⑥ $T(n) \; \varepsilon \; O(2^n)$
⑦ $T(n) \; \varepsilon \; \Omega(1)$
8. $T(n) \; \varepsilon \; \Theta(n)$
⑨ $T(n) \; \varepsilon \; O(n!)$

(g)The following is an execution of Project 3. Your task is to draw the Red Black Tree that exists after these commands are executed. Please be specific and show all of the content of each node. In addition, draw the nil node and show what nodes point to it. Color nodes red with a circle around the node. Black nodes have no circle. (7 Points)

f 4 =
4
a 5 =
5
h a f + =
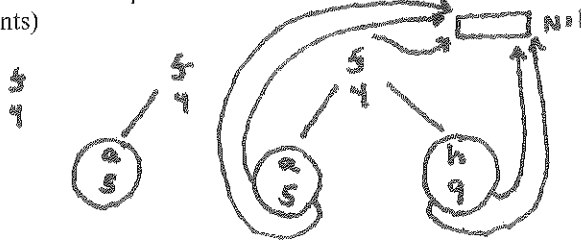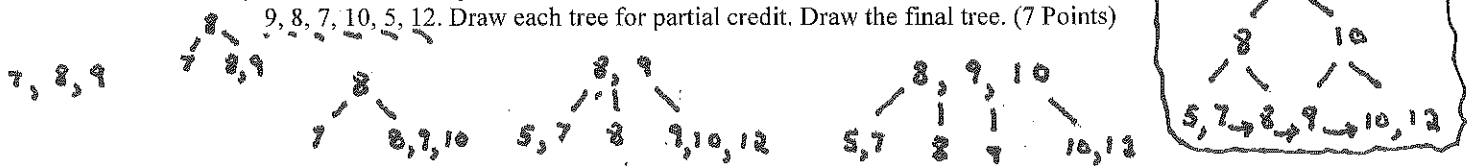
B Trees (21 points)

10.   a) Insert the following numbers into a B-Tree with a minimum of 2.
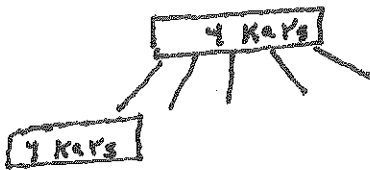   9, 8, 7, 6, 5, 12, 2   Draw the final tree. (7 Points)

b) Insert the following numbers into a B+ Tree with a minimum of 1.
9, 8, 7, 10, 5, 12. Draw each tree for partial credit. Draw the final tree. (7 Points)

c)   Consider a B-Tree with a minimum of 2. What is the exact maximum number of keys such a tree could hold if the tree were of height 1? __24__ (7 Points)
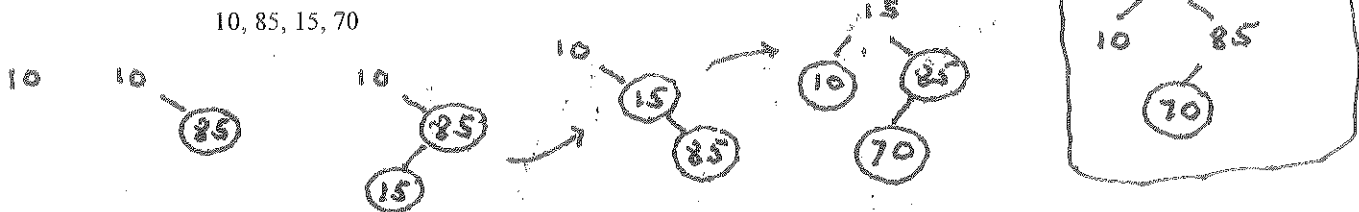
$4 + 5 \cdot 4 = 24$

Red Black Trees (8 points)

11. Red Black Trees

(a) Insert the following numbers, one by one, into a Red-Black Tree. Show the tree after each insertion. Draw RED nodes with a circle or a label 'R'. (5 points)

10, 85, 15, 70

(b) When trying to analyze the run time complexity of an inorder traversal of a Red Black tree, one should consider the worst case, average case and best case separately. Each case may have a different run time performance. Circle TRUE or FALSE. (1 point)

(c) When trying to analyze the run time complexity of a Red Black tree insertion, one needs to consider the worst case, average case and best case. Circle TRUE or FALSE (1 point)

(d) What is the worst case runtime complexity of a Binary Search Tree insert operation? This is not a Red Black tree. Use Big Theta notation. (1 points) $\Theta(N)$

Graph Algorithms (26 points)

See the attached graphs, G_1, G_2.

12. (a) What is the shortest path from the start node 0 to node 4 in the graph G_1? Your path must be a list of ordered pairs. (1 point) $(0,1), (1,2), (2,4)$

(b) Show the list of nodes that would be visited by a breadth first search in the graph G_2. We are starting from vertex 0. (1 point) $0, 1, \{2,5,6\}, \{3,4\}$ MANY ORDERINGS EXIST

(c) Draw the contents of the distance array for each iteration of Dijkstra's Algorithm as it works on the graph sketched here. The initial state is given. **Mark the node to be selected next to the left of the array (note how 0 is marked to the left of the first array.)** Fill in each array cell working downward. That is, complete the first column of arrays before the second column of arrays. (11 Points)
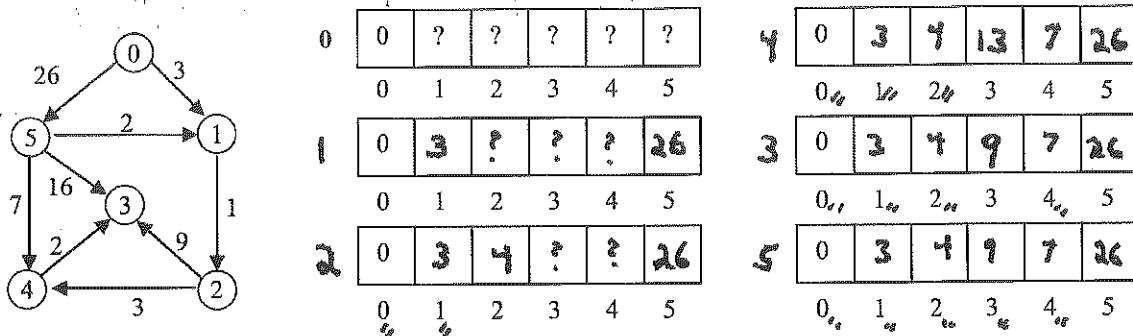
(d) After Dijkstra is complete it also collects parent pointers for each node in the graph. Complete the chart below showing the parent of each node as computed by Dijkstra. (3 point)
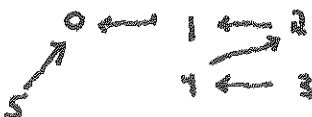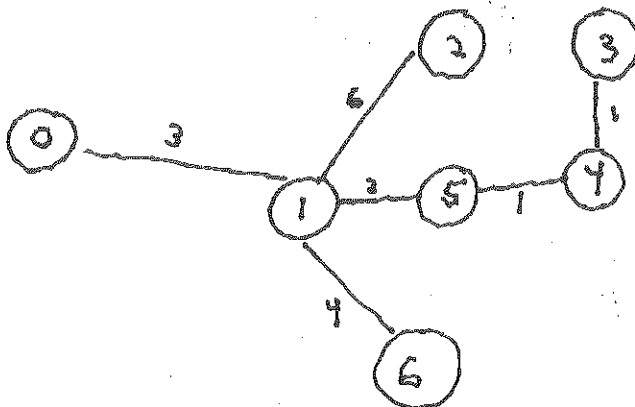
Key 10

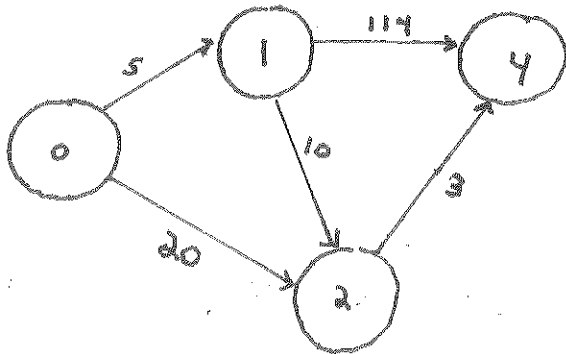| Node | Parent |
======	========
0	Nil
1	0
2	1
3	4
4	2
5	0

(e) Assume that you have routines to mark a vertex, process a vertex and place a vertex in a queue. You also have a routine that allows you to access each unmarked neighbor of a vertex. Write an algorithm that determines if a path exists starting at vertex v and ending at vertex w. Your algorithm will perform a breadth first traversal on the graph like the one above in question (12 c.). You need not code this in Java. But the algorithm should be clear to the reader. (8 points) The signature of the method isPath looks like the following:

// returns true if and only if there is a path from vertex v to vertex w.
boolean isPath(int v, int w);

```
boolean found = false
Queue q = new queue
mark (v)
q.enqueue(v)
while ! q.isEmpty() and not found {
    u = q.dequeue
    if u == w then found = true
    else for each unmarked neighbor X of u do
        mark X
        q.enqueue(X)
    }
    return found
```

(f) Suppose you were to run Prim's algorithm on graph G_2. Draw an MST that Prim would compute. There may be several answers. Draw one MST. (2 points)

6.

G₂