

Name Key

Andrew ID Key

Total points = 133. Score will be a percentage of 133.

Tracing Lists and Trees (30 points)

1. You will be asked to show the exact output of the following program. (20 Points)

```
class Pair {
    private int x;
    private int y;
    public Pair(int x, int y) {
        this.x = x;
        this.y = y;
    }
    public void setX(int x) {
        this.x = x;
    }
    public void setY(int y) {
        this.y = y;
    }
    public int getX() {
        return x;
    }
    public int getY() {
        return y;
    }
    @Override
    public String toString() {
        return "x == " + x + " y == " + y + " ";
    }
}
class Node {
    private Node previous;
    private Pair data;
    private Node next;

    public Node (Node p, int x, int y, Node n) {
        previous = p;
        data = new Pair(x,y);
        next = n;
    }
    public Node getNext() {
        return next;
    }
    public Node getPrevious() {
        return previous;
    }
}
```



```
}
public Pair getData() {
    return data;
}
public void setData(Pair data) {
    this.data = data;
}
public void setNext(Node next) {
    this.next = next;
}
public void setPrevious(Node previous) {
    this.previous = previous;
}
@Override
public String toString() {
    return data.toString();
}
}
public class List {
    int first = 1;
    int second = 2;
    Node head = null;
    Node last = null;
    void add() {
        if (head == null) {
            head = new Node(null, first, second, null);
            last = head;
        }
        else {
            Node p = head;
            Node q = p;
            while (p != null) {
                q = p;
                p = p.getNext();
            }
            last = new Node(q, first, second, null);
            q.setNext(last);
        }
        first = second + first;
        second = second + first;
    }

    public String toString() {
        Node p = last;
        String temp = "";
        while (p != null) {
```



```

    temp = temp + p.toString();
    p = p.getPrevious();
}
return "List: " + temp;
}
// In need of a pre-condition. Question 1.g
public void remove() {
    if(head.getNext() == null) {
        head = null;
        last = null;
    }
    else {
        head = head.getNext();
        head.setPrevious(null);
    }
}
}
public static void main(String[] args) {
    List myList = new List();
    myList.add();
    myList.add();
    myList.add();
    System.out.println(myList); // Question 1.a
    myList.remove();
    System.out.println(myList); // Question 1.b
    myList.remove();
    System.out.println(myList); // Question 1.c
    myList.remove();
    System.out.println(myList); // Question 1.d
}
}

```

- 1.(a) Show the output of the code marked (1.a). LIST: x==8 y==13 x==3 y==5 x==1 y==2 3  
Points
- 1.(b) Show the output of the code marked (1.b). LIST: x==8 y==13 x==3 y==5 3  
Points
- 1.(c) Show the output of the code marked (1.c). LIST: x==8 y==13 4  
Points
- 1.(d) Show the output of the code marked (1.d). LIST: 2  
Point
- 1.(e) What is the Big Theta value for the add method?  $\Theta(\log N)$  2  
Point
- 1.(f) Is it correct to say that the method toString() of the list class runs in  $O(n^2)$ ?  
Circle YES or NO (2 pt.)



1.(g) Provide a pre-condition for the remove() method in the List class. Be as precise as possible. (4 Points)

```
// Precondition: head != null  
public void remove{...}
```

2. Study the execution of the following program. Note that the logic may differ from what we have studied. Questions appear below.

```
class TreeNode {  
    public int data;  
    public TreeNode lc;  
    public TreeNode rc;  
    public TreeNode parent;  
  
    public TreeNode(TreeNode lc, int x, TreeNode rc, TreeNode parent) {  
        this.lc = lc;  
        this.data = x;  
        this.rc = rc;  
        this.parent = parent;  
    }  
}  
public class ExamTree {  
  
    public TreeNode root;  
    public TreeNode last;  
  
    public ExamTree() {  
        root = null;  
    }  
    public void add(int x) {  
        if (root == null) {  
            root = new TreeNode(null, x, null, null);  
            last = root;  
        }  
        else {  
            TreeNode t = root;  
            TreeNode q = t;  
            while(t != null) {  
                if(x >= t.data) {  
                    q = t;  
                    t = t.rc;  
                }  
                else {  
                    q = t;  
                    t = t.lc;  
                }  
            } // end while  
            if(x >= q.data) {  
                q.rc = new TreeNode(null, x, null, q);  
                last = q.rc;  
            }  
            else {  
                q.lc = new TreeNode(null, x, null, q);  
            }  
        }  
    }  
}
```

Page 12



```
        last = q.lc;
    }
}

public void traversal(TreeNode 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 visit() {
    TreeNode ptr = last;
    while( ptr != null) {
        System.out.println(ptr.data);
        ptr = ptr.parent;
    }
}

public static void main(String[] args) {
    ExamTree st = new ExamTree();
    st.add(6);
    st.add(8);
    st.add(4);
    st.traversal(); // Question 2 (a)
    st.add(7);
    st.add(8);
    st.add(1);
    st.traversal(); // Question 2 (b)
    st.visit(); // Question 2 (c)
}
}
```

2.(a) What is the output from the traversal method labelled Question 2 (a)? 1 Point

4 6 8

2.(b) What is the output from the traversal method labelled Question 2 (b)? 1 Point

1 4 6 7 8 8

2.(c) What is the output from the traversal method labelled Question 2 (c)? 2 Points

1 4 6

2.(d) Suppose this tree is perfectly balanced (unlike the tree created by the insertions shown above). What will be the runtime complexity for the visit() method if the tree has n nodes? Your answer will be in Big Theta notation. 2 Points

$\Theta(\log N)$



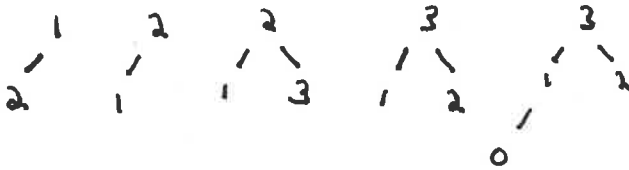
2.(e) Does the traversal method named `traverse()` have a worst case that differs from its best case? Circle YES or **NO** 1 Point

2.(f) What is the worst case Big Theta value for the `add` method?  $\Theta(N)$  1 Point

2.(g) What is the best case Big Theta value for the `add` method?  $\Theta(1)$  2 Point  
Heaps (12 points)

3) Insert the following 6 numbers into a max heap. Draw a new tree for each heap insertion. (4 Points)

1, 2, 3, 0



4) What is the height of the tree that you drew in question 3? (A single node in a tree gives a height of 0.)  
(2 Points) 2

5) Perform exactly one `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. That is, show each step clearly. (3 Points)



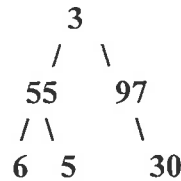
6) Consider the following min heap implemented in an array. It is not quite correct. To make it a proper min heap exactly one swap must occur. What two numbers (child and parent) need to be swapped in order to make this a min heap? (3 points). PLACE CHECK MARKS NEXT TO THE TWO NUMBERS THAT NEED TO BE SWAPPED. [12,40,81,70,71,82,80]

✓      ✓



Trees (16 points)

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



- (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)

3 55 6 5 97 30

- (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 55 5 3 97 30

- (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)

3 55 97 6 5 30

- (d) In general, if a binary tree (at most two children per node) is perfectly balanced (unlike the tree pictured above) and complete with  $n$  nodes, how many leaves will the tree have? (2 points)  $(n+1)/2$  Note, this tree has a perfectly flat bottom. We need the total number of leaves in terms of  $n$ . This is an exact answer, not Big O.

- (e) In general, if a binary tree (at most two children per node) is perfectly balanced (unlike the tree pictured above) and complete with height  $h$ , how many internal nodes (non-leaves) will the tree have? (1 point)

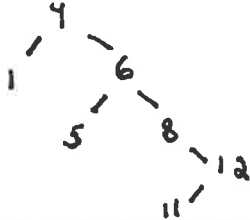
$2^h - 1$

How many leaves will this tree have? (1 point)  $2^h$

Note, this tree has a perfectly flat bottom. These are exact answers, not Big O.



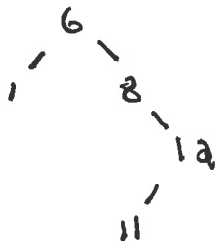
8. (a) Insert the following numbers into a Binary Search Tree. Draw the tree after all insertions are complete. (1 Point)  
4, 1, 6, 8, 12, 11, 5



(b) Delete the 4 from the final tree that you drew in 8 (a). We are following the “go right once, left hard” rule. Draw this final tree. (2 Point)



(c) Delete 5 from the final tree that you drew in question 8 (b). Again, we are following the “go right once, left hard” rule. Draw this final tree. (2 Points)







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  $X = \{4,6,9,3\}$  and  $k = 11$ . Is there a subset of  $X$  which sums to  $k$ ?  
No Yes/No (1 point)

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

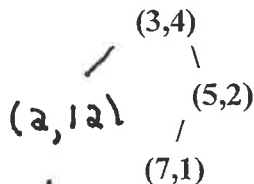
1. a problem that is impossible to solve.
2. a problem that has been proven to take exponential time to solve.
3. a problem that has been proven to take factorial time to solve.
4. an optimization problem.
- ⑤ a decision problem.

(c) Suppose Alice sends the integer  $K$  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 eavesdropper could read the message  $M$  if the eavesdropper could (circle the one best option) (1 Point)

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

(d) Recall that a modular inverse of an integer  $b$  mod  $m$  is the integer  $b^{-1}$  such that  $(b * b^{-1}) \bmod m = 1$ . What is the modular inverse of  $4 \bmod 7$ ? 2 (2 Points)

(e) The following points, in a standard  $(x,y)$  coordinate plane, have been added to a 2-d tree.  $(3,4)$ ,  $(5,2)$ ,  $(7,1)$ . The 2-d tree appears as follows:



Add the point  $(2,12)$  to this 2-d tree. Redraw the tree with this new point added. The first point,  $(3,4)$ , breaks the plane vertically. (2 points)



- (f) Consider the 2-d tree that you created, with the addition of (2,12), in (e). Suppose that we performed a nearest neighbor search for the point (1,4). Which point(s) in the tree need to be examined?

(3, 7), (2, 12)  
(2 Points)

- (g) In Project 3 we wrote a Red Black binary search tree. Suppose we are doing a lookup for a course name in the Red Black Tree. Let  $T(n)$  be the number of operations required to do the lookup. 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.  $T(n) \in \Theta(\text{Log}(n))$
- 2.  $T(n) \in O(2^n)$
- 3.  $T(n) \in \Theta(n)$
- 4.  $T(n) \in O(\text{Log}(n))$
- 5.  $T(n) \in O(3^n)$
- 6.  $T(n) \in O(n^{1/2})$
- 7.  $T(n) \in \Omega(n^{1/2})$
- 8.  $T(n) \in O(n!)$
- 9.  $T(n) \in \Omega(n)$
- 10.  $T(n) \in \Omega(n^2)$
- 11.  $T(n) \in O(n)$

- (h) The following is a data file for Project 3. Note the course Span1 that is taken by Alice and Carol. The purpose of the Red-black tree was to associate an integer with each course name. What integer will be assigned to Carol's Span1 in the Red-Black tree of Project 3? Note: the first course we see is assigned the number 0. (4 Points)

1

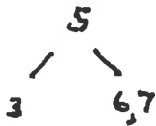
```
Alice Calc1 Span1 Philo2 Logic2
Bob Calc1 Philo2 Hist4 Logic2
Carol Hist4 Span1 Eng4
```

- (i) Consider the graph generated by the Project 3 dataset in question 9 (h). We are referring to the previous question. How many edges will the node labelled "Eng4" have? (3 Points) 2

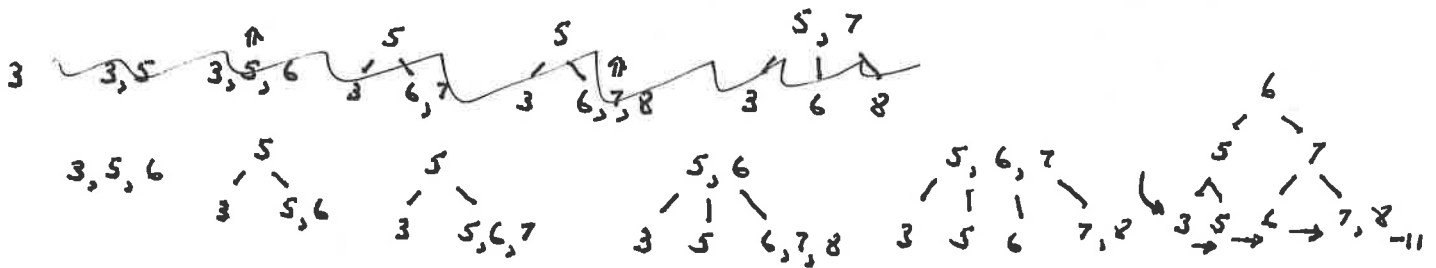


**B and B+ Trees (21 points)**

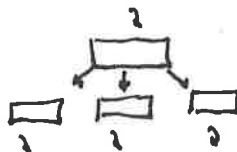
10. (a) Insert the following numbers into a B-Tree with a minimum of 1.  
3,5,6,7 Draw the final tree. (7 Points)



- (b) Insert the following numbers into a B+ Tree with a minimum of 1:  
3, 5, 6, 7, 8. Draw each tree for partial credit. Draw the final tree. (7  
Points) Note: This is a B+ tree and not a B-Tree.



- (c) Consider a B-Tree with a minimum of 1. What is the exact maximum  
number of  
keys such a tree could hold if the tree were of height 1? 8  
(7 Points)



$2 + 2 + 2 + 2 = 8$



**Red Black Trees (8 points)**

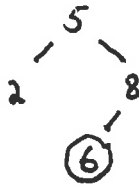
11. Red Black Trees : Please be careful with your colors

(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'. (4 points)

5, 2, 8



(b) Add the number 6 to the Red-Black tree in question 11(a). Draw RED nodes with a circle or a label 'R'. (4 points)



**Graph Representations (9 points)**

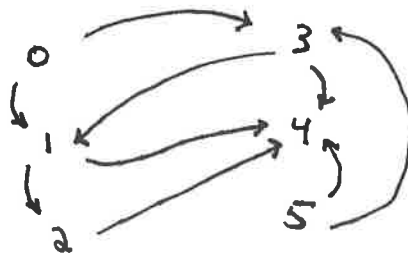
Consider a directed graph  $G_1$ . The graph is represented by an adjacency matrix  $m$ . If there is an edge from  $i$  to  $j$  then  $m[i,j] = \text{true}$ .

Matrix  $m$

vertex	0	1	2	3	4	5
0		true		true		
1			true		true	
2					true	
3		true			true	
4						
5				true	true	

$G_1$

12. (a) Draw the graph  $G_1$  with circles and edges. (2 Points)



Key





12. (b) Suppose that we decide to represent the graph  $G_1$  with an adjacency list rather than an adjacency matrix. Draw a picture of what that representation would look like.

(3 Points)

```

0 - 3 - 1
1 - 4 - 2
2 - 4
3 - 4 - 1
4 - 11
5 - 4 - 3
    
```

12. (c) Given the graph  $G_1$  and starting at vertex 5, what are the first 3 vertices visited by a breadth first traversal of the graph. (We are counting vertex 5 as one of the first 3).

(3 Points). 5, 4, 3 or 5, 3, 4

12. (d) Given the graph  $G_1$  and starting at vertex 5, with a depth first traversal, we will visit all of the vertices in the graph. Circle TRUE or FALSE.

(1 Point).

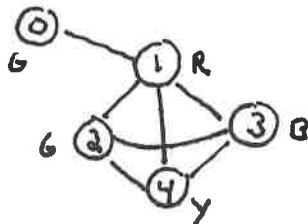
Graph Coloring (9 points)

13. Color the graph  $G_2$  with as few colors as possible. Use convenient colors - such as red, blue, yellow, green, cyan, etc. We need to see a drawing of the graph as well as the color of each node.

A matrix representation of the graph  $G_2$  appears here:

vertex	0	1	2	3	4
0		true			
1	true		true	true	true
2		true		true	true
3		true	true		true
4		true	true	true	

13 (a) Draw the graph  $G_2$  with a proper coloring(5 points):



13 (b) How many colors are required for an optimal coloring? 4 (4 Points)



Big O (4 Points)

14. I have just arrived by plane at Pittsburgh International Airport and I have forgotten where I have parked my car. There are  $n$  cars in the parking lot. I decide to look for my car with a simple serial search, examining, one by one, each and every car in the lot. This algorithm could be analyzed by considering its best case, average case, and worst case. Circle each of the following answers that are mathematically correct. (4 Points)

- a. In the worst case, the algorithm is Big Omega( $n$ ).
- b. In the worst case, the algorithm is Big Theta( $n^2$ ).
- c. In the worst case, the algorithm is Big Omega( $n^2$ ).
- d. In the best case, the algorithm is Big O( $n!$ ).
- e. In the best case, the algorithm is Big O( $2^n$ ).
- f. In the best case, the algorithm is Big Theta( $n$ ).

Reduce Key (4 Points)

15. Given the following minheap, how many swaps will occur if 10 is reduced to 1 ?

[3,10,15,12,14,18,20]

Place your answer here: 1

