95-771 Data Structures and Algorithms
Spring 2025

Midterm Exam
Carnegie Mellon University

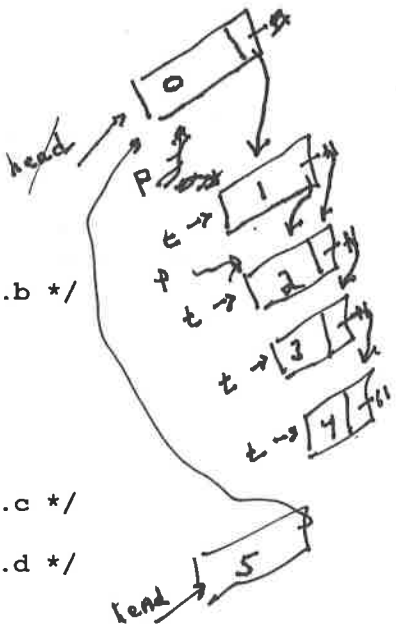
Name Key **Andrew ID** Key

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)

```
public class Node {

    public int data;
    public Node next;
    public Node(int value, Node ptr) {
        data = value;
        next = ptr;
    }
    public String toString() {                    /* 1.a */
        String temp = Integer.toString(data);
        if(next == null) {
            temp = temp + "--| ";
            return temp;
        }
        else {  // make a recursive call
            return temp + "," + next.toString() ;
        }
    }
    public static void main(String args[]) {

        Node head = new Node(0,null);
        System.out.println("Line 1: " + head);   /* 1.b */
        Node p = head;
        for (int i = 1; i <5; i++) {
            Node t = new Node(i, null);
            p.next = t;
            p = p.next;
        }
        System.out.println("Line 2: " + head);   /* 1.c */
        head = new Node(5,head);
        System.out.println("Line 3: " + head);   /* 1.d */
    }
}
```



For question 1, assume n is the number of nodes on the list.

1.(a) What is the Big Theta for the toString() method marked with (1.a)? $\Theta(n)$ (3 pts)
1.(b) Show the output of the code marked (1.b). Line 1: 0 --| (3 pts)
1.(c) Show the output of the code marked (1.c). Line 2: 0, 1, 2, 3, 4 --| (3 pts)
1.(d) Show the output of the code marked (1.d). Line 3: 5, 0, 1, 2, 3, 4 --| (3 pts)
1.(e) Is it correct to say that the method toString() of the Node class runs in $\Omega(2^n)$?
Circle True or False (3 pt.)
1.(f) Is it correct to say that the method toString of the Node class runs in $\Omega(1)$?
Circle True or False (3 pt.)

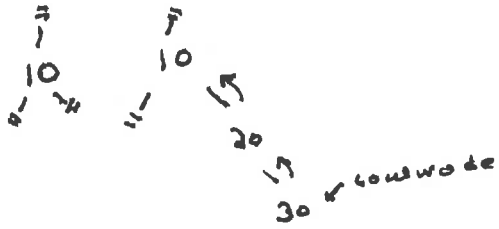
KEY

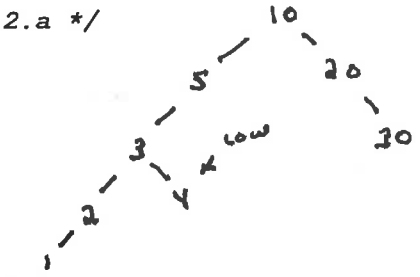
2. Study the execution of the following program. This is not the same program that appeared on a previous exam. Five questions appear below. (10 points):

```
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;
            }
        }
    }
```

Key

```
    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);
        System.out.println("(2.a)");
        st.traversal2();                /* Question 2.a */

        st.add(5);
        st.add(3);
        st.add(2);
        st.add(1);
        st.add(4);

        System.out.println("(2.b)");
        st.traversal2();                /* Question 2.b */

        System.out.println("(2.c)");
        st.traversal();                 /* Question 2.c */

    }
}
```





In the following questions, you may ignore new lines. Your answers will appear on one line.

2.(a) What will the program display by the call to traversal2() marked Question 2.a? (2 Points)

30 20 10

2.(b) What will the program display by the call to traversal2() marked Question 2.b? (2 Points)

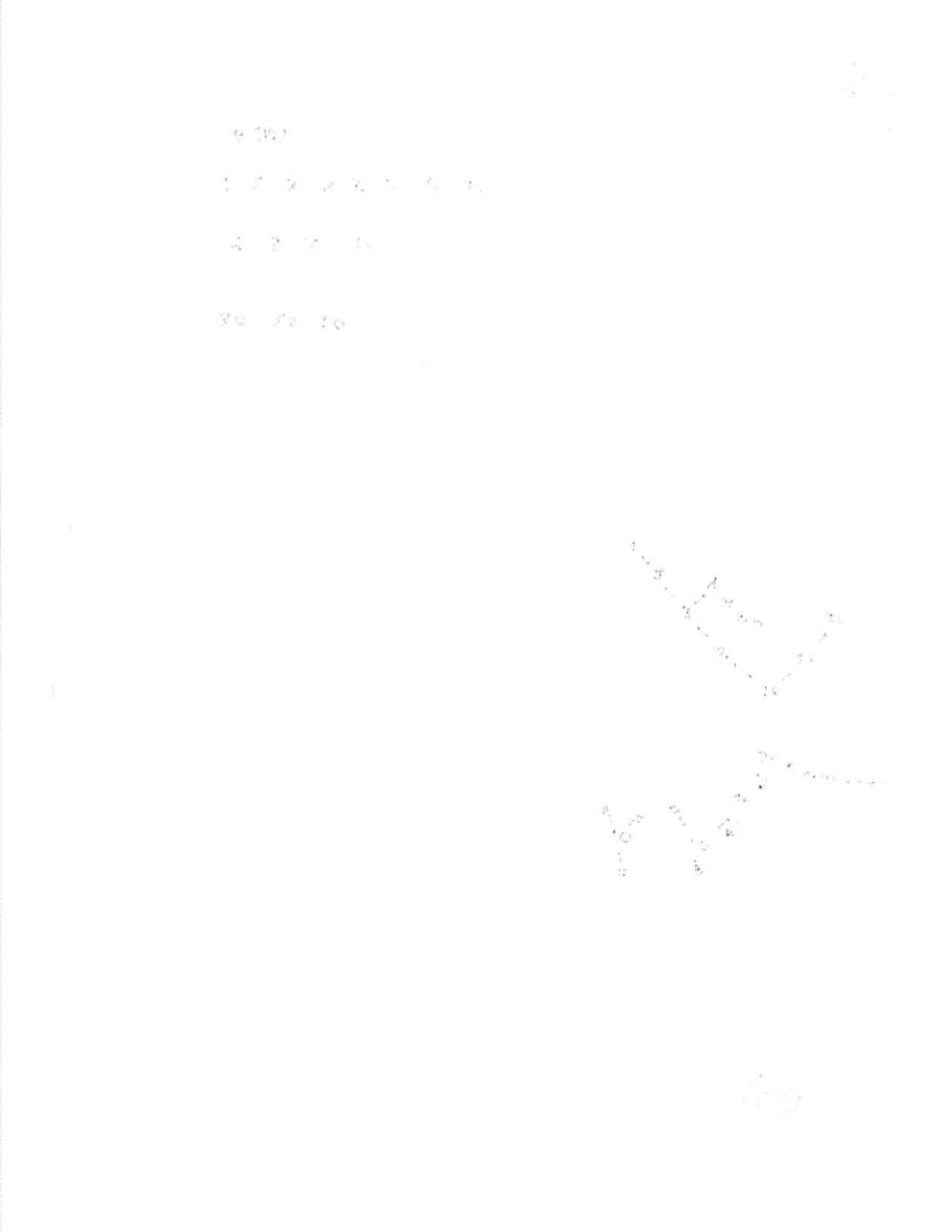
4 3 5 10

2.(c) What will the program display by the call to traversal() marked Question 2.c? (2 Points)

1 2 3 4 5 10 20 30

2.(d) What is the worst case Big Theta value for traversal2(). Our only assumption is that the tree holds n nodes in total. $\Theta(N)$ (1 point)

Key



2.(e) Suppose we built a CLR complete tree and it was perfectly balanced (unlike the tree created above). Suppose too that the tree held n nodes in total. Provide a Big Theta value for traversal(). $\Theta(N)$
(1 point)

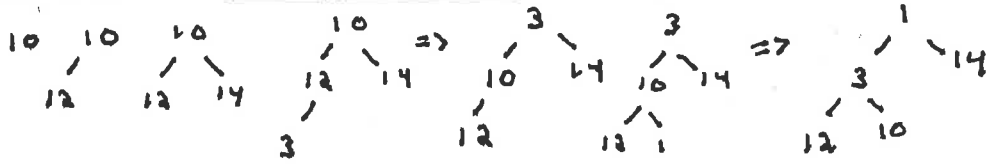
2.(f) Suppose we built a CLR complete tree and it was perfectly balanced (unlike the tree created above). Suppose too that the tree held n nodes in total. Provide a Big Theta value for traversal2(). $\Theta(\log N)$
(1 point)

2.(g) Suppose we built a CLR complete tree and it was perfectly balanced (unlike the tree created above). Suppose too that the tree held n nodes in total. Is it correct to say that traversal2() is O(2^n)?
(1 point) Circle TRUE or False.

Heaps (12 points)

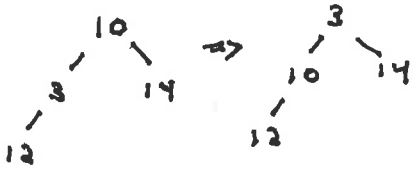
3) Insert the following 5 numbers into a min heap. Draw a new tree for each heap insertion. (4 Points)

10, 12, 14, 3, 1



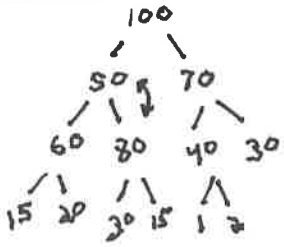
4) What is the height of the tree that you drew in question 3? (A single node in a tree gives a height of 0.)
(2 Points) 2

5) Perform exactly one deleteMin() operation on the heap that you drew in question 3. Draw the resulting tree. With a careful drawing, make it clear to the reader what is going on. (3 Points)



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

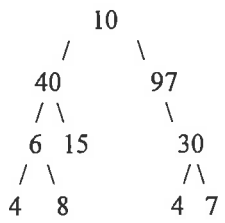
100
50 ✓
70
60
80 ✓
40
30
15
20
30
15
1
2



Binary Trees (16 points)

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

Key



(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)

10 40 6 4 8 15 97 30 4 7

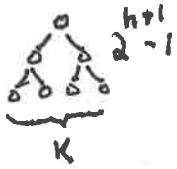
(b) List the data that would be accessed by an in-order traversal on the given tree by writing out the values in the nodes as they would be accessed, separated by commas. (2 points)

4 6 8 40 15 10 97 4 30 7

(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)

10 40 97 6 15 30 4 8 4 7

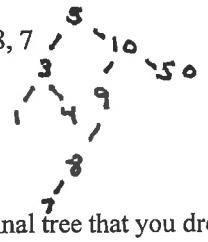
(d) In general, if a binary (at most two children per node) tree is perfectly balanced (unlike the tree pictured here) and complete with height h, how many nodes, in terms of h, will the tree have? (2 points) $2^{h+1}-1$ Note, this tree has a perfectly flat bottom. We need the total number of nodes in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



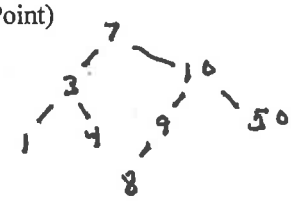
(e) In general, if a binary tree is perfectly balanced (unlike the tree pictured here) and complete with exactly k leaves. What is the number of nodes (in terms of k) in this tree? (2 points)

$K+K-1 = 2K-1$ Note, this tree has a perfectly flat bottom. This is an exact answer, not Big O.

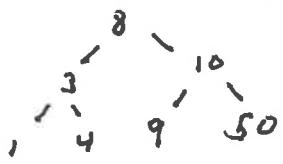
8. (a) Insert he following numbers into a Binary Search Tree. Draw the tree after all insertions are complete. (1 Point)
5, 10, 50, 3, 1, 4, 9, 8, 7



(b) Delete 5 from the final tree that you drew in 8 (a). We are following the "Go Right, Left Hard" rule. Draw this final tree. (2 Point)



(c) Delete 7 from the final tree that you drew in 8 (b). This is the tree that resulted from the deletion of 5 in 8 (b). We are following the "Go Right, Left Hard" rule. Draw this final tree. (2 Point)



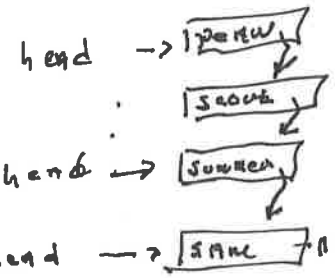
key

Project Questions (20 points)

(9) Recall your work with linked lists from Project 1, 2-D trees from Project 2 and Red Black trees from Project 3.

(a) Given the following LinkedList program, complete the method named display(). The method named display() will print every data value in the list. The first value on the list is printed first, then the second, and so on. Place your Java programming logic in the method named display(). (4 Points)

```
class Node {
        String data;
        Node link;
        public Node(String value, Node next) {
                data = value;
                link = next;
        }
}
public class LinkedList {

    public static void display(Node ptr) {
        while (ptr != null) {
            System.out.println(ptr.data);
            ptr = ptr.next;
        }                    LINK

    }
    public static void main(String[] args) {
        Node head = new Node("Sam",null);
        head = new Node("Summer",head);
        head = new Node("Scout", head);
        head = new Node("Peanut", head);
        display(head);
    }
}
```



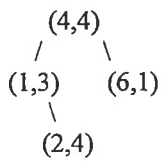
(b) What is an appropriate pre-condition for the method named display() in question 9 (a) ? (1 Point)

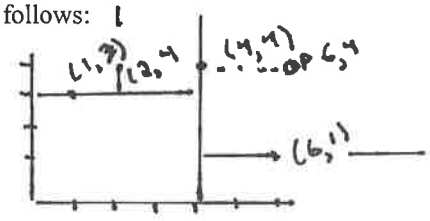
Pointer LIST IS NULL terminated

(c) What will be the output of the completed program in 9 (a)? This is the program with the completed display() method. (2 Points)

Peanut Scout Summer Sam

(d) The following points, in a standard (x,y) coordinate plane, have been added to a 2-d tree. (4,4), (6,1), (1,3), (2,4). The 2-d tree appears as follows:

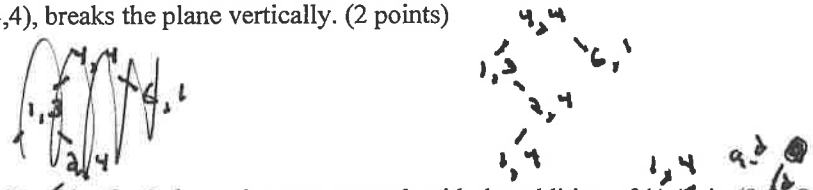




key

Key

Add the point (1,4) to this 2-d tree. Redraw the tree with this new point added. The first point, (4,4), breaks the plane vertically. (2 points)



(e) Consider the 2-d tree that you created, with the addition of (4,4), in (9.1). Suppose that we performed a nearest neighbor search for the point (6,4). Which points in the tree need to be examined? {(4,4) (6,1)} (2 Points)

(f) In Project 3 we wrote a Red Black binary search tree. Suppose we are doing an insert of a course 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.) (5 Points)

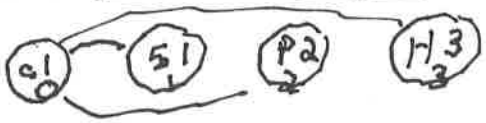
1 Pt OFF per error

5 pts OFF max.

1. $T(n) \varepsilon O(1)$
2. $T(n) \varepsilon \Omega(1)$
3. $T(n) \varepsilon \Omega(n^2)$
4. $T(n) \varepsilon O(n)$
5. $T(n) \varepsilon \Theta(\text{Logn})$
6. $T(n) \varepsilon O(2^n)$
7. $T(n) \varepsilon \Theta(n)$
8. $T(n) \varepsilon O(n!)$
9. $T(n) \varepsilon O(\text{Log}(n))$

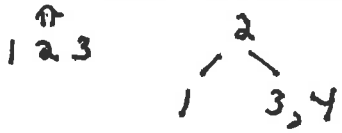
(g)The following is a data file for Project 3. Note the course Hist3 that is taken by Sue. The purpose of the Red-black tree was to maintain an integer with each course name. What integer will be assigned to Sue's Hist3 in the Red-Black tree of Project 3? The course numbers begin at 0. (4 Points) 3

Amy Calc1 Span1 Philo2 Hist3
Bill Calc1 Philo2 Hist4
Sue Calc1 Hist3

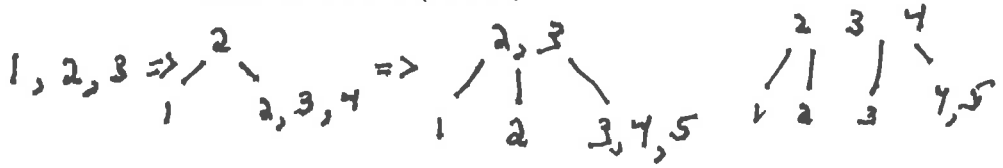


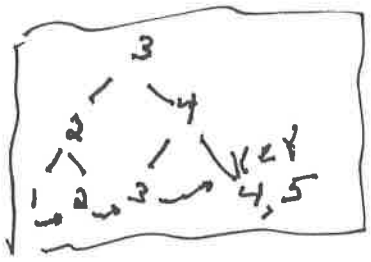
B Trees (21 points)

10. (a) Insert the following numbers into a B-Tree with a minimum of 1. max == 2
1,2,3,4 Draw the final tree. (7 Points)



(b) Insert the following numbers into a B+ Tree with a minimum of 1.
1,2,3,4,5. Note, this is a B+ tree. Draw each tree for partial credit.
Draw the final tree. (7 Points)

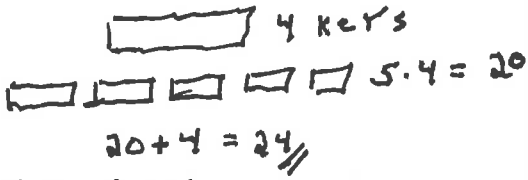




Key

Key

(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)

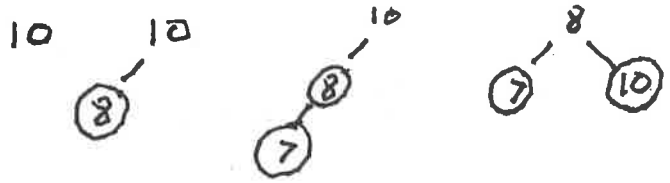


Red Black Trees (8 points)

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'. (8 points)

10, 8, 7



Graph Algorithms (26 points)

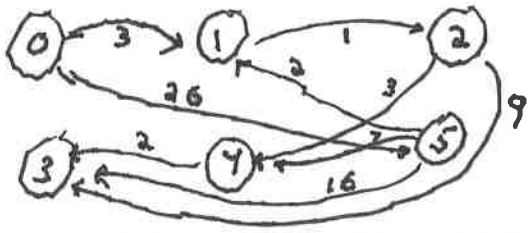
Consider the weighted, directed graph G_1 . The graph is represented by an adjacency matrix m. If there is an edge from i to j with weight k then m[i,j] = k.

Matrix m

vertex	0	1	2	3	4	5
0		3				26
1			1			
2				9	3	
3						
4				2		
5		2		16	7	

G_1

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



12. (b) What is the shortest path from the start node 0 to node 3 in the graph G_1? Your path must be a list of ordered pairs. (2 points) (0,1), (1,2), (2,4), (4,3)

3 1 3 2

Key

Key

Consider the undirected graph G_2. The graph is represented by an adjacency matrix n. If vertex i shares an edge with vertex j then n[i,j] = T.

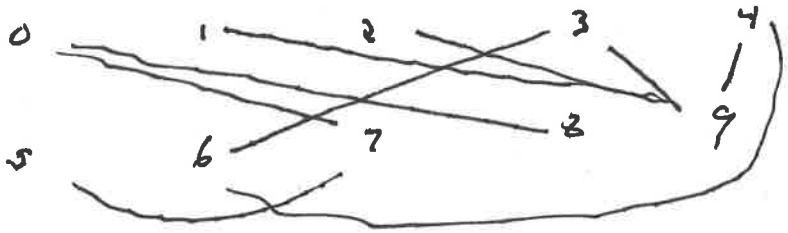
Matrix n

Vertex	0	1	2	3	4	5	6	7	8	9
0								T	T	
1										T
2										T
3							T			T
4							T			T
5								T		
6				T	T					
7	T					T				
8	T									
9		T	T	T	T					

G_2

12. (c) Draw the G_2 graph with circles and edges. (2 points)

4 connects to 9 and 6



12. (d) 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. (3 points) {0, 8, 7}, {5},

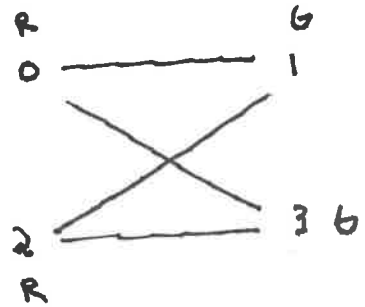
Consider the undirected graph G_3. The graph is represented by an adjacency matrix o. If vertex i shares an edge with vertex j then o[i,j] = T.

Martrix o

vertex	0	1	2	3
0		T		T
1	T		T	
2		T		T
3	T		T	

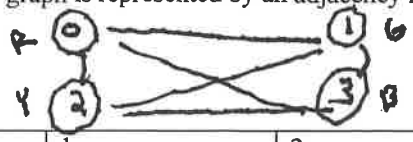
G_3

12. (e) What is the minimum number of colors that we could color G_3 with? (3 Points) 2



Key

Consider the undirected graph G_4. The graph is represented by an adjacency matrix p. If vertex i shares an edge with vertex j then p[i,j] = T.



Martrix p

vertex	0	1	2	3
0		T	T	T
1	T		T	T
2	T	T		T
3	T	T	T	

G_4

12. (f) What is the minimum number of colors that we could color G_4 with ? (3 Points) 4

Consider the undirected graph G_5. The graph is represented by an adjacency matrix q. If vertex i shares an edge with vertex j then q[i,j] = T.

Martrix q

Vertex	0	1	2	3	4
0		T		T	
1	T		T		T
2		T		T	T
3	T		T		
4		T	T		

G_5

12. (g) What is the minimum number of colors that we could color G_5 with ? (3 Points) 3

12. (h) Suppose we are working with the graph G_5. Write a method that displays each immediate neighbor of vertex v. You may assume that we have 5 nodes – 0 through 4. This will be written in error free Java. The algorithm should be clear to the reader. (8 points)

For example, if v == 1 then the method will display 0, 2, 4.

Complete the body of the method displayNeighbors(). Place your Java programming logic inside the braces of displayNeighbors().

Also, include any preconditions that are appropriate.

```
// PRE: V >= 0 AND V <= 4
// prints each immediate neighbor of vertex v
public void displayNeighbors( int v) {
    FOR (INT i = 0; i <= 4; i++) {
        IF (g[v][i]) SYSTEM.OUT.PRINT(i);
    }
}
```