95-771 Data Structures and Algorithms
Spring 2022

Midterm Exam
Carnegie Mellon University

Name _____ **Andrew ID** Key

Total points = 13 2. Score will be a percentage of 13 2.

Tracing Lists and Trees (28 points)

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

```
package s22midterm;

class Node {
    private String data;
    private Node next;
    private Node prev;

    public String getData() {
        return data;
    }
    public void setData(String data) {
        this.data = data;
    }

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

    public Node getPrev() { return prev; }
    public void setPrev(Node prev) {
        this.prev = prev;
    }

    public Node(Node prev, String data, Node next) {
        this.data = data;
        this.next = next;
        this.prev = prev;
    }
}
class List {
    Node head;
    public List() {
        head = null;
    }
    /* (1.a) Big theta */
    public void add(String x) {
        if(head == null) {
            head = new Node(null, x, null);
        }
        else {
            Node temp = new Node(null,x,head);
            head.setPrev(temp);
            head = temp;
        }
    }
```

Key

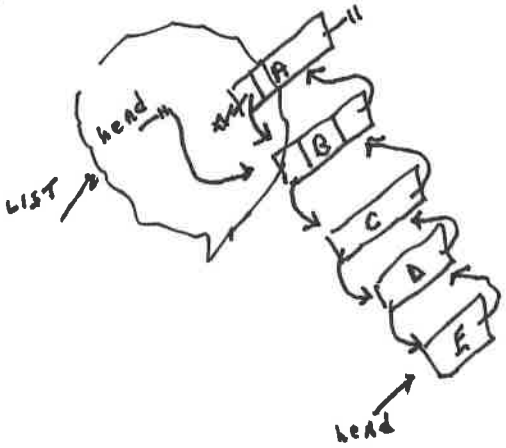
```
    public boolean isEmpty () {
        return head == null;
    }

    // (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 String traverse() {
        String result = "";
        Node p = head;
        Node q = p;
        while (p != null) {
            q = p;
            p = p.getNext();
        }
        while(q != null) {
            result = result + q.getData() + " ";
            q = q.getPrev();
        }
        return result;
    }
}

public class DSMidterm {

    public static void main(String[] args) {
        List list = new List();
        list.add("A");
        list.add("B");
        list.add("C");
        list.add("D");
        list.add("E");
        System.out.println(list);     /* 1.d */
        System.out.println(list.traverse());   /* 1.e */
        List list2 = list;
        System.out.println(list2);    /* 1.f */
        System.out.println(list2.traverse());  /* 1.g */

    }
}
```



1.(a) Give the Big Theta of the code marked (1.a) (add). Θ(1) (2 pts)
1.(b) Give the Big Theta of the code marked (1.b) (toString). Θ(N) (2 pts)
1.(c) Give the Big Theta of the code marked (1.c) (traverse). Θ(N) (1 pts)
1.(d) Show the output of the code marked (1.d). E D C B A (1 pts)
1.(e) Show the output of the code marked (1.e). A B C D E (4 Points)
1.(f) Show the output of the code marked (1.f). E D C B A (4 Points)
1.(g) Show the output of the code marked (1.g). A B C D E (4 Points)

Key

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

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

2. Study the execution of the following program. 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;
            }
```

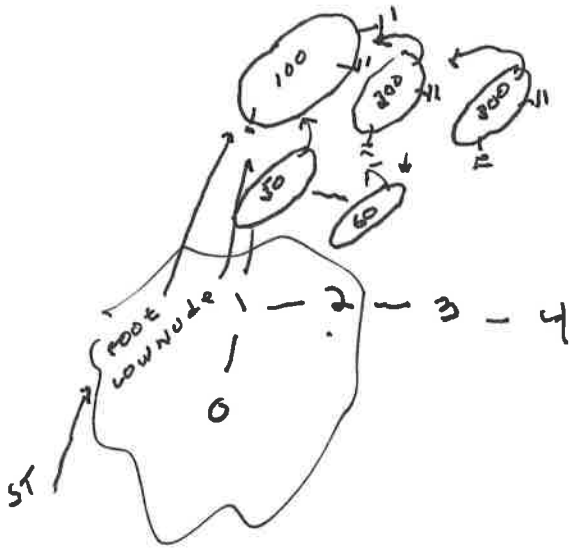

Key

Key

```
        }
    }
    public void traversal(Node r) {
        if(r == null) return;
        if(r.lc != null)traversal(r.lc);
        if(r.rc != null)traversal(r.rc);
        System.out.println(r.data);
    }
    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(100);
        st.add(200);
        st.add(300);
        st.add(50);
        st.add(60);
        System.out.println("(2.a)");
        st.traversal2();
        st.add(1);
        st.add(0);
        st.add(2);
        st.add(3);
        st.add(4);
        System.out.println("(2.b");
        st.traversal2();
        System.out.println("(2.c");
        st.traversal();

    }
}
```



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

60 50 100

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

4 3 2 1 50 100

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

0 4 3 2 1 60 50 300 200 100

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

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(). $\Theta(N)$ (2 points)

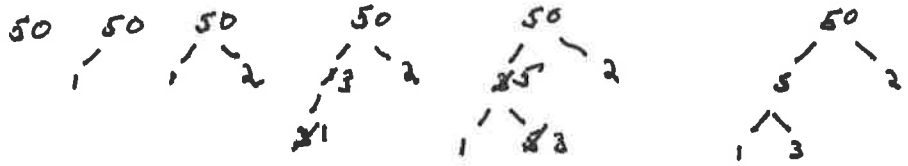
Key

Key

Heaps (12 points)

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

50, 1, 2, 3, 5

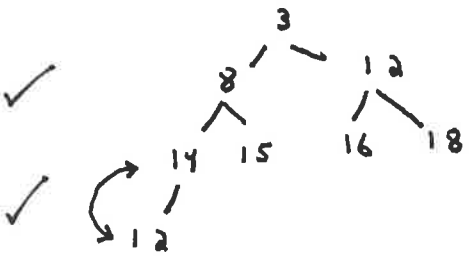


4) What is the height of the tree that you drew in question 3? (A single node in a tree gives a height of 0.) (2 Points) 2

5) Perform exactly 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 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.

3
8
12 ✓
14
15
16
18
12 ✓

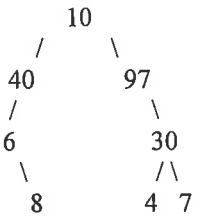


Key

Key

Binary 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)

10, 40, 6, 8, 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)

6, 8, 40, 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)

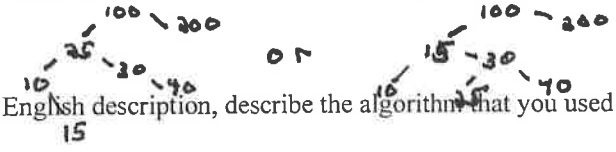
10, 40, 97, 6, 30, 8, 4, 7

(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 height (in terms of k) of this tree? (2 points) $\log_2 k$ 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)
100, 200, 20, 10, 15, 30, 25, 40

(b) Delete 20 from the final tree that you drew in 8 (a). Draw this final tree. (1 Point)



(c) Using an English description, describe the algorithm that you used for this deletion. (3 Point)

Go right once and LEFT hand for P

replace node to be deleted with P's value

delete P

Key

key

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 = {100, 9, 2, 105, 3, 7, 101} and **k** = 18. Is there a subset of X which sums to k?
Yes/No (1 point)

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

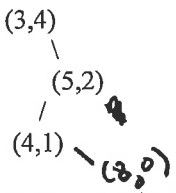
1. an optimization problem.
2. a problem that is impossible to solve.
3. a problem that has been proven to take exponential time to solve.
4. a problem that has been proven to take factorial time to solve.
5. a decision problem.

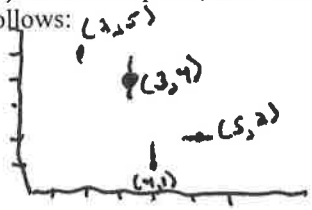
(c) Suppose Alice sends a message (K) to Bob. K is computed using Bob's Merkle-Hellman public key combined with the message M. The central idea behind Merkle-Hellman is that a potential eaves dropper could read the message M if the eaves dropper could (circle the one best option) (1 Points)

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 $(b^* b^{-1}) \bmod m=1$. What is the modular inverse of 3 mod 11? 4 (2 Points) $3 \cdot 4 = 12 \quad 12 \bmod 11 = 1$

(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:





Add the point (8,0) 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 (8,0), in (e). Suppose that we performed a nearest neighbor search for the point (2,5). Which points in the tree need to be examined? (3,4), (5,2) (2 Points)

key

Key

(g) 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.) (4 Points)

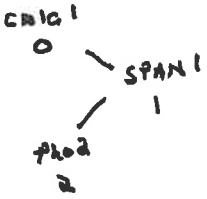
1. $T(n) \varepsilon O(1)$
2. $T(n) \varepsilon \Omega(n^2)$
3. $T(n) \varepsilon O(n)$
4. $T(n) \varepsilon \Theta(\text{Log}n)$
5. $T(n) \varepsilon O(2^n)$
6. $T(n) \varepsilon \Theta(n)$
7. $T(n) \varepsilon O(n!)$
8. $T(n) \varepsilon \Omega(1)$
9. $T(n) \varepsilon O(\text{Log}(n))$

3, 4, 5

7, 8, 9

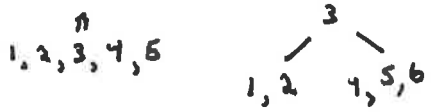
(h)The following is a data file for Project 3. Note the course Philo2 that is taken by Bill. The purpose of the Red-black tree was to maintain an integer with each course name. What integer will be assigned to Bill's Philo2 in the Red-Black tree of Project 3? (7 Points) 2

Amy Calc1 Span1 Philo2 Hist3
Bill Calc1 Philo2 Hist4

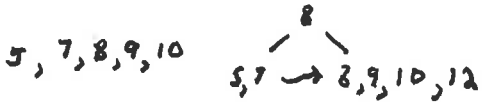


B Trees (21 points)

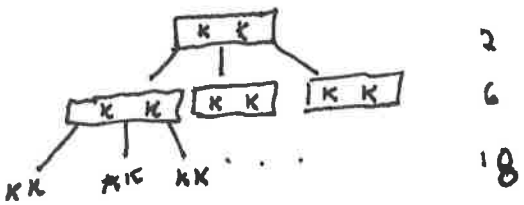
10. (a) Insert the following numbers into a B-Tree with a minimum of 2.
1,2,3,4,5,6 Draw the final tree. (7 Points)



(b) Insert the following numbers into a B+ Tree with a minimum of 2.
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 1. What is the exact maximum number of keys such a tree could hold if the tree were of height 2? 26 (7 Points)



Key

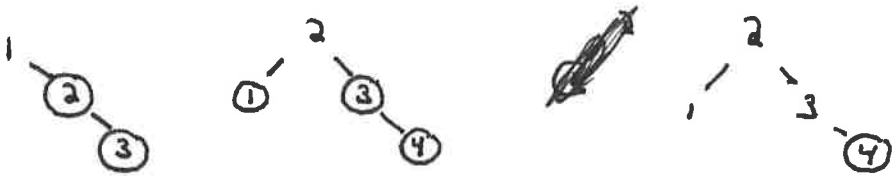
Key

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)

1,2,3,4



Graph Algorithms (26 points)

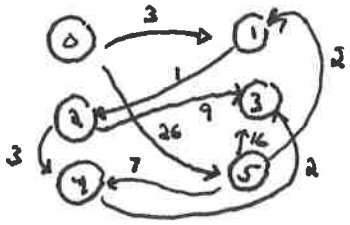
Consider the weighted, directed graph G_1. The graph is represented by an adjacency matrix m. If there is an edge from i to j with weight k then m[i,j] = k.

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)

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

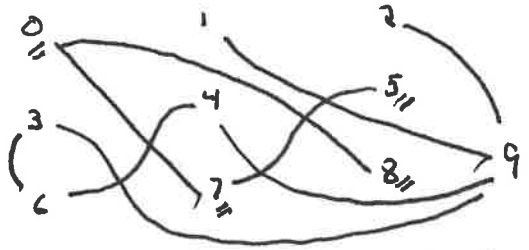
Key

Key

	0	1	2	3	4	5	6	7	8	9
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)



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, 7, 8, 5

ANY ORDER

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

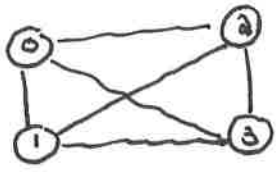
12. (e) What is the minimum number of colors that we could color G_3 with? (3 Points) 2

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



Key

KEY

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	T
2		T		T	T
3	T	T	T		
4		T	T		

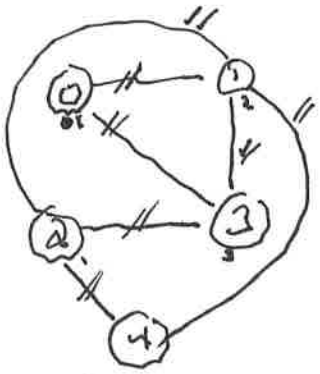
G_5

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.

The signature of the method displayNeighbors() looks like the following:




```
// prints each immediate neighbor of vertex v
public void displayNeighbors( int v);
```

```
Public void displayNeighbors (int v) {
    For (i = 0; i <= 4; i++) {
        If (G5[v, i] == true) System.out.Print(i)
    }
}
```


KEY