95-771 Data Structures and Algorithms
Spring 2017

Midterm Exam
Carnegie Mellon University

Name TA

Andrew ID _____

Total points = 120. Score will be a percentage of 120.

Coding Lists and Trees (28 points)

1. You will be asked to show the exact output of the following program. The program makes two calls on display(). Show the output from each call to display().

```
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;
    public List() {
      head = null;
    }
    public void add(int x) {
      head = new Node(x,head);
    }
    public int front() { return head.getData(); }
    public String toString() {
      Node v = head;
      String s = "";
      while(v != null) {
        s = s + v.getData() + " ";
        v = v.getNext();
      }
      return s;
    }
  }
```

0 1 2

```
class TreeNode {
   List data;
   TreeNode lc,rc;

   public TreeNode(List data, TreeNode lc, TreeNode rc) {
      this.data = data;
      this.lc = lc;
      this.rc = rc;
   }

   public List getData() {
      return data;
   }

   public TreeNode getRc() {
      return rc;
   }

   public TreeNode getLc() {
      return lc;
   }

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

   public void setLc(TreeNode lc) {
      this.lc = lc;
   }

   public void setRc(TreeNode rc) {
      this.rc = rc;
   }

}
class BinarySearchTree {

TreeNode root;

BinarySearchTree()
{
   root = null;
}

public void addNode(int n)
{
   List l = new List();
   int j = 0;
   while(j < n) {
      l.add(j);
      j = j + 1;
   }
   if(root == null)
   {
      root = new TreeNode(l,null,null);
   }
```

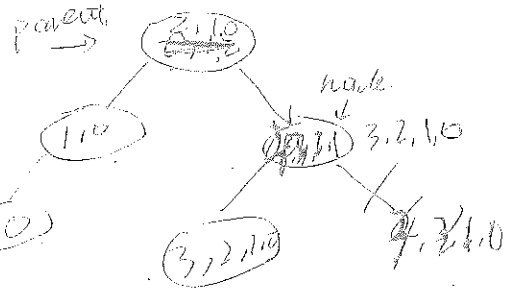
0, 1, 2

```
        else
        {
           TreeNode node = root;
           TreeNode parent = null;
           while(node!=null)
           {
              parent = node;
              if(node.getData().front() + 1 < n)
                 node = node.getRc();
              else
                 node = node.getLc();
           }
           int data = parent.getData().front() + 1;
           if(data < n)
           {
              parent.setRc(new TreeNode(l,null,null));
           }
           else
           {
              parent.setLc(new TreeNode(l,null,null));
           }
        }
     }
     public void display(TreeNode r) {
        if(r != null) {
           display(r.getRc());
           display(r.getLc());
           System.out.println(r.getData());
        }
     }
     public void display() {
        display(root);
     }
  }
public class DSMidterm {

   public static void main(String[] args) {

      BinarySearchTree bst = new BinarySearchTree();
      bst.addNode(3);
      bst.display();     // answer question 1 a.
      bst.addNode(2);
      bst.addNode(4);
      bst.addNode(4);
      bst.addNode(1);
      bst.display();     // answer question 1 b.

   }
  }
```



1.a What will the program display at bst.display() marked 1.a? ~~2,1,0~~ 2,1,0 (6 pts)

1.b What will the program display at bst.display() marked 1.b? 3,2,1,0 (10 pts)

3,2,1,0
0
1,0
2,1,0

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

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

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

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

   public Node root;
   public int ctr;

   public SimpleTree() {
      root = null;
      ctr = 1;
   }

   private Node add(Node t){
      if (t == null) {
         ctr = ctr + 2;
         return new Node(null,ctr,null);
      }
      t.lc = add(t.lc);
      t.rc = add(t.rc);
      return t;
   }

   public void add() {
      if (root == null) {
         ctr = ctr + 2 ;
         root = new Node(null,ctr,null);
      }
      else {
         add(root);
      }
   }

   public void traversal(Node t) {

      if(t != null) {
         System.out.print(t.data + ":");
         traversal(t.lc);    // NOTE: Left child first
         traversal(t.rc);    // Note Right child second
      }
```

3:5:7:

3:5:9:11:7:13:15:

```
  }
  public void traversal() {
    traversal(root);
  }

  public static void main(String[] args) {
    SimpleTree st = new SimpleTree();
    st.add();
    st.add();

    // Question 2.a
    st.traversal();
    System.out.println();

    SimpleTree coolTree = new SimpleTree();
    coolTree.add();
    coolTree.add();
    coolTree.add();

    // Question 2.b
    coolTree.traversal();

  }
}
```

2.a What will the program display at the traversal marked Question 2.a? (5 Points)

3 : 5 : 7 :

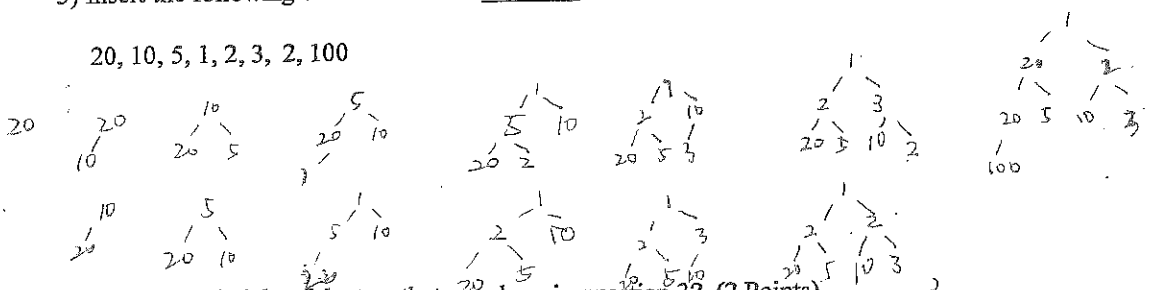
2.b What will the program display at the traversal marked Question 2.b? (5 Points)

3 : 5 : 9 : 11 : 7 : 13 : 15 :

Heaps (12 points)

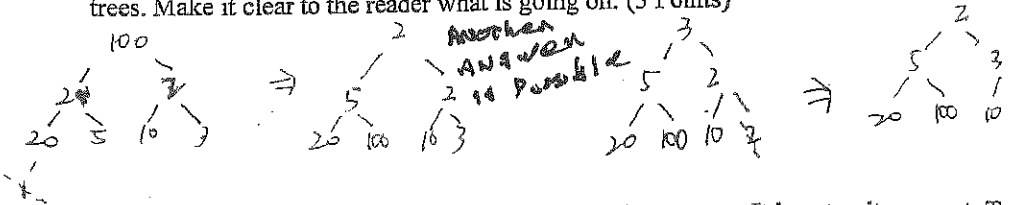
3) Insert the following 8 numbers into a min heap. Draw a new tree for each heap insertion. (4 Points)

20, 10, 5, 1, 2, 3, 2, 100



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

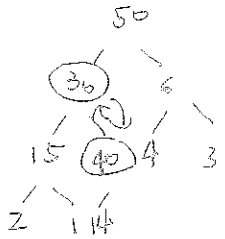
5) Perform exactly two deleteMin() operations on the heap that you drew in question 3. Draw the resulting trees. Make it clear to the reader what is going on. (3 Points)



6) Consider the following 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

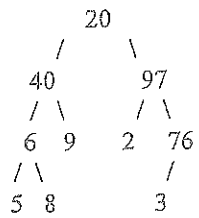
to make this a max heap? (3 points). PLACE CHECK MARKS NEXT TO THE TWO NUMBERS THAT NEED TO BE SWAPPED.

50
30
6
15
40
4
3
2
1
14



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)

20, 40, 6, 5, 8, 9, 97, 2, 76, 3

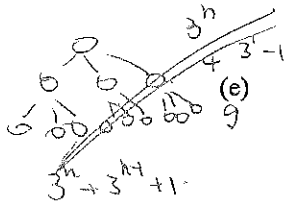
(b) List the data that would be accessed by an in-order traversal on the given tree by writing out the values in the nodes as they would be accessed, separated by commas. (2 points)

5, 6, 8, 40, 9, 20, 2, 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)

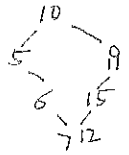
20, 40, 97, 6, 9, 2, 76, 5, 8, 3

(d) In general, if a ternary (at most three children per node) tree is perfectly balanced (unlike the tree pictured here) and complete with height h, how many leaves, in terms of h, will the tree have? (2 points) 3^h Note, this tree has a perfectly flat bottom.

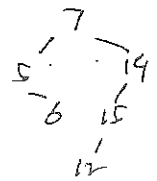


(e) In general, if a ternary tree is perfectly balanced (unlike the tree pictured here) and complete with exactly k leaves. What is the height (in terms of k) of this tree? (2 points) $\log_3 k$ Note, this tree has a perfectly flat bottom.

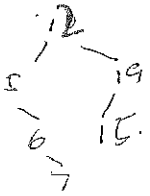
8. (a) Insert he following numbers into a Binary Search Tree. Draw the tree after all insertions are complete. (3 Points)
10, 5, 6, 7, 19, 15, 12



(b) Delete 10 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.



OR



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} and k = 18. Is there a subset of X which sums to k?
_____ Yes/No (2 points)

(b) The type of problem you were asked to solve in question 9 (a) is (Circle one answer): (2 Points)

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.

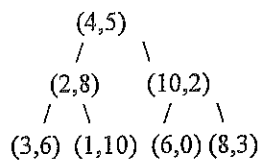
(c) Suppose Alice sends a message (K) to Bob. K is computed using Bob's Merkle-Hellman public key combined with the message M. The central idea behind Merkle-Hellman is that a potential eaves dropper could read the message M if the eaves dropper could (circle the one best option) (2 Points)

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 $(b * b^{-1}) \bmod m = 1$.
What is the modular inverse of 2 mod 13? __7__ (1 Points)

$2 \cdot 2^{-1} \bmod 13 = 1$

(e) Consider the following 2-d tree:



A reverse level order would visit the nodes in what order ? List the nodes. (2 Points)

(3,6) (1,10) (6,0) (8,3) (2,8) (10,2) (4,5)

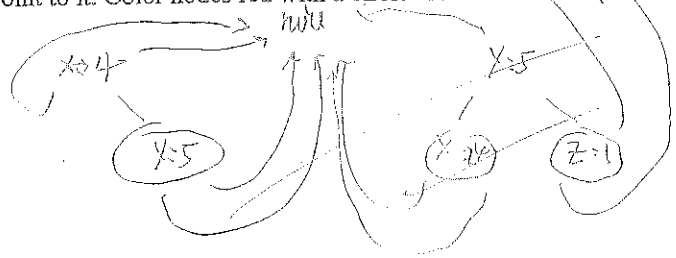
(f) In Project 3 we wrote a Red Black binary search tree. Suppose we are doing an insert of a variable name into a Red Black Tree. Let T(n) be the number of operations required to do the insert. In the worst case, which of the following are true about T(n)? Circle all of those that are true. (You may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

1, 4, 5, 6, 7.

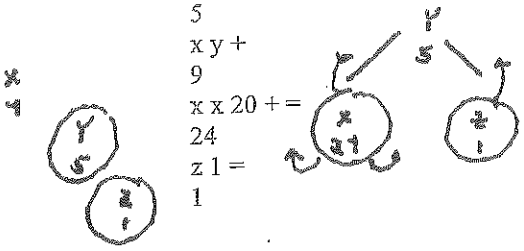
1. T(n) ε O(LogN)
2. T(n) ε O(1)
3. T(n) ε $\Omega(N^2)$
4. T(n) ε O(N)
5. T(n) ε Θ(LogN)
6. T(n) ε $O(2^n)$

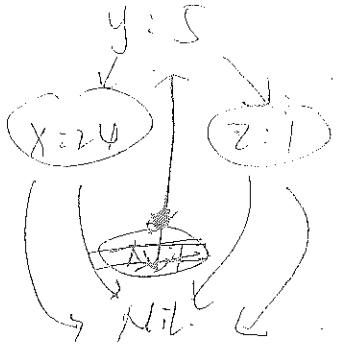
7. $T(n) \in \Omega(1)$
8. $T(n) \in \Theta(N)$

(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. (7 Points)



x 4 =
4
y 5 =
5
x y +
9
x x 20 + =
24
z 1 =
1

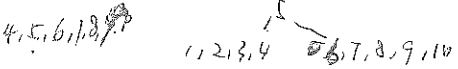




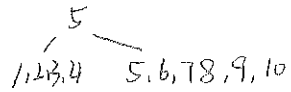
B Trees (21 points)

10. a) Insert the following numbers into a B-Tree with a minimum of 4.
1,2,3,4,5,6,7,8,9,10. Draw the final tree. (7 Points)

1,2,3,4,5,6,7,8,9,10



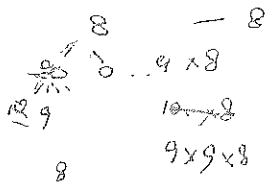
b) Insert the following numbers into a B+ Tree with a minimum of 4.
1,2,3,4,5,6,7, 8, 9, 10. Draw the final tree. (7 Points)

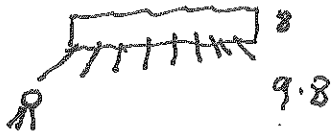


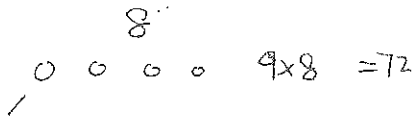
$9^0 \cdot 8 + 9^1 \cdot 8 + 9^2 \cdot 8$

$8 + 72 + 648 = 728$

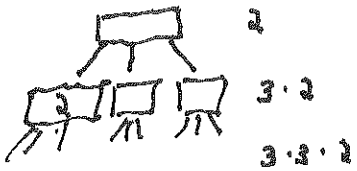
c) Consider a B-Tree with a minimum of 4. What is the exact maximum number of keys such a tree could hold if the tree were of height 2? 728 (7 Points)



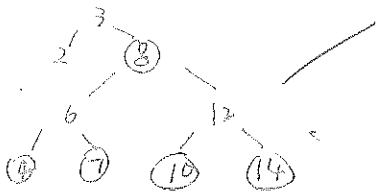




9 x 9 x 9 = 729



$3^0 \cdot 2 + 3^1 \cdot 2 + 3^2 \cdot 2$

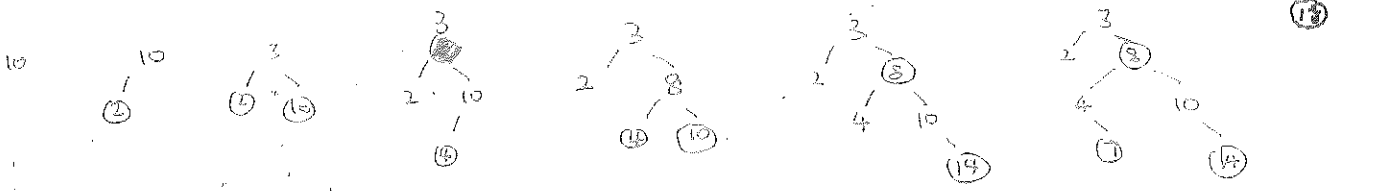


Red Black Trees (8 points)

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)

10,2,3,4,8,14,7,6,12



(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) What is the best-case runtime complexity of a Red Black Tree insert operation? Use Big Theta notation. (1 points) Log N

(d) What is the best-case runtime complexity of a Binary Search Tree insert operation? Use Big Theta notation. (1 points) ~~Log N~~ 1

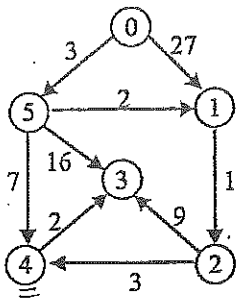
Graph Algorithms (26 points)

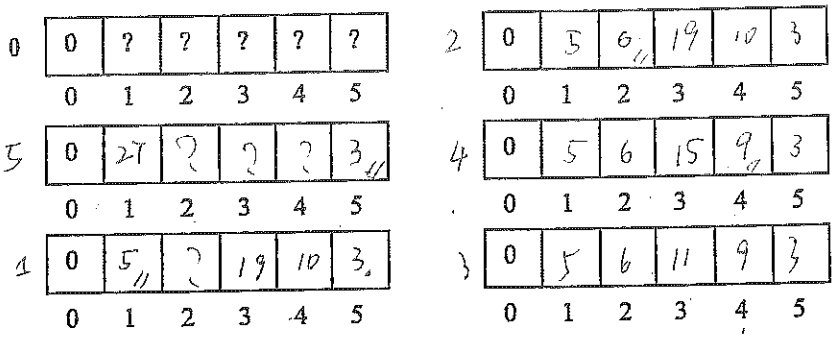
See the attached graphs, G_1, G_2 and G_3.

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), (1,4)}.

(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, 6, 5, 3, 2, 1, 4.

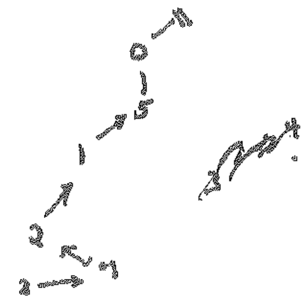
(c) Draw the contents of the distance array for each iteration of Dijkstra's Algorithm as it works on the graph sketched here. The initial state is given. Mark the node to be selected next to the left of the array (note how 0 is marked to the left of the first array.) Fill in each array cell working downward. That is, complete the first column of arrays before the second column of arrays. (6 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)

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



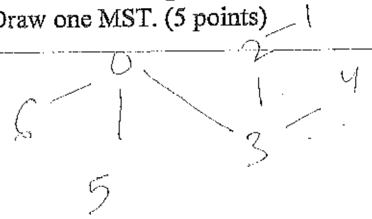
(e) Assume that you have routines to mark a vertex, process a vertex and place a vertex in a queue. You also have a routine that allows you to access each unmarked neighbor of a vertex. Write an algorithm that performs a breadth first traversal on the graph like the one above. You will need to process each vertex visited by the breadth first traversal. That is, call process(x) for each vertex visited. The input to your algorithm is vertex v. (3 points)

BFS (V)
Mark V
Progress V
Engrave V

while (Que is empty) {
d = queue.poll;
fore each unmarked child of d:
mark child;
access child;
Add child to queue;

(f)If a directed graph with |V| vertices and |E| edges is represented by an adjacency matrix, what will be the space complexity of such a representation? (1 Points) $|V|^2$

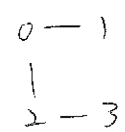
(g)Suppose you were to run Prim's algorithm on graph G_2. Draw an MST that Prim would compute. There are several answers. Draw one MST. (5 points)



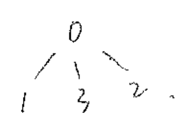
One possible answer.

(h)Show the MST that Prim would compute on G_3. Also, show the paths that Dijkstra would compute on G_3. In each case, start work from node 0. Draw the two results. State what you conclude about Dijkstra and Prim. Do they compute the same result ? (3 points)

Prim:



Dijkstra:



NOT the same

(i)Below is a recursive version of Floyd Warshall. Note: the real Floyd Warshall uses dynamic programming. Your problem is to show the output of this program. Note that it is too time consuming to trace the recursion. You will need to think about what Floyd Warshall does and compute the correct values yourself. The values -1 are used to signify that this column or row is unused. Java uses 0 indexing in arrays but Floyd Warshall describes the first node in the graph as node 1. The values 1000 are used to signify infinity. Show the exact output as computed by this program. (3 Points)

```
public class FloydWarshallRecursive {

  public static int d(int w[][],int i, int j, int k){

    if ( k == 0) return w[i][j];

    else return Math.min(d(w,i,j,k-1),(d(w,i,k,k-1) + d(w,k,j,k-1)));
  }

  public static void main(String[] args) {

    int cost[][] = { {-1, -1, -1, -1}, {-1, 0, 1000, 4}, {-1, 1, 0, 3}, {-1, 1000, 5, 0} };

    System.out.println(d(cost,3,2,3));
    System.out.println(d(cost,3,1,3));
    System.out.println(d(cost,2,3,3));

  }
}
```

	1	2	3
-1	-1	-1	-1
1	-1	0	1000	4
2	-1	1	0	3
3	-1	1000	5	0