95-771 Data Structures and Algorithms
Spring 2023

Midterm Exam
Carnegie Mellon University

Name Key **Andrew ID** key@andrew.cmu.edu

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

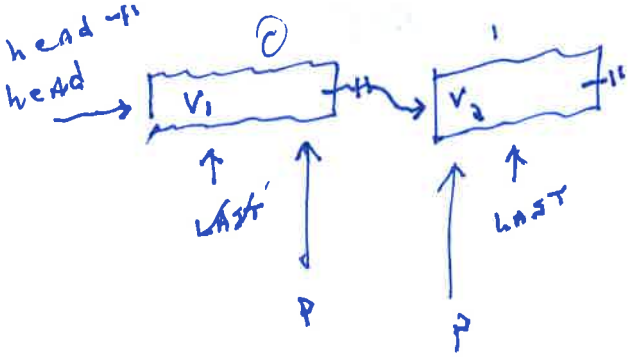
Tracing Lists and Trees (30 points)

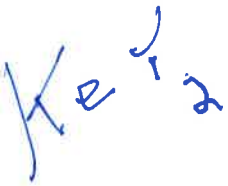
1. You will be asked to show the exact output of the following program. (20 Points)

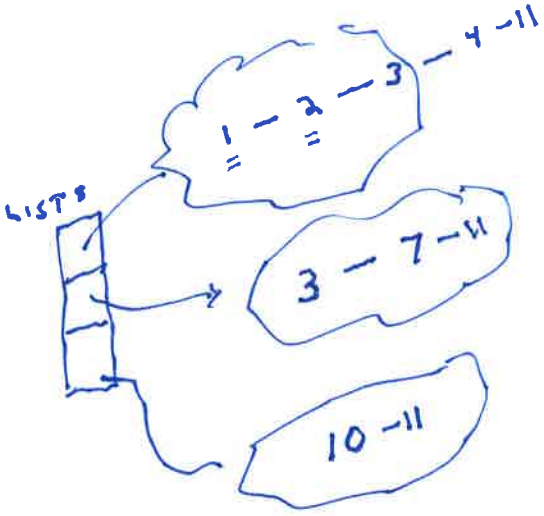
```
package org.example;

class Node {
   private int data;
   private Node next;
   public int getData() {
      return data;
   }
   public void setData(int data) {
      this.data = data;
   }
   public Node getNext() {
      return next;
   }
   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 last;
   int ctr;
   public List() {
      head = null;
      last = null;
      ctr = 0;
   }
   public int getCtr() {
      return ctr;
   }
```

Key 1

```
   public void add(int value) {
      if (head == null) {
         head = new Node(value,null);
         last = head;
      }
      else {
         last.setNext(new Node(value,null));
         last = last.getNext();
      }
      ctr++;
   }
// number of nodes > i
public int get(int i) {
      Node p = head;
      for (int m = 0; m < i; m++) {
        p = p.getNext();
      }
      return p.getData();
   }
public String toString() {
      Node v = head;
      String s = "";
      while(v != null) {
         s = s + v.getData() + " ";
         v = v.getNext();
      }
      return s;
   }
}

public class DSAMidterm {

   public static void main(String[] args) {

      List[] lists = new List[3];
      lists[0] = new List();
      for(int i = 1; i <= 4; i++) {
         lists[0].add(i);
      }
      lists[1] = new List();
      for(int i = 0; i < 2; i++) {
         lists[1].add(lists[0].get(2 * i) + lists[0].get(2*i+1));
      }
      lists[2] = new List();
```







```
        lists[2].add(lists[1].get(0) + lists[1].get(1));

        // Question 1.a
        System.out.println(lists[0]);

        // Question 1.b
        System.out.println(lists[1]);

        // Question 1.c
        System.out.println(lists[2]);

        // Question 1.d
        for (int t = 0; t < lists.length; t++) {
            System.out.print(lists[t].getCtr() + " ");
        }
    }
}
```

1 2 3 4 get 3

3 7

10

4 2 1

1.(a) Show the output of the code marked (1.a). 1 2 3 4 2 Points
1.(b) Show the output of the code marked (1.b). 3 7 2 Points
1.(c) Show the output of the code marked (1.c). 10 2 Points
1.(d) Show the output of the code marked (1.d). 4 2 1 2 Points

1.(e) Is it correct to say that the method add() of the list class runs in O(n!) ?
Circle True or False (1 pt.)

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

1.(g) Provide a pre-condition for the get method in the List class. (4 Points)

```
// Precondition: NUMBER OF NODES > i ≥ 0
public int get(int i) {...}
```

ctr > i

1.(h) Write a recursive routine for the class List above. This routine will print the list values in reverse order. It is fine if you also decide to write two methods that, used together, give the caller a convenient way to display the values on the list in reverse order. You must write valid Java and you may not use a stack. (6 Points)

```
void PRINT_rev ( Node P ) {
    IF (P == NULL) return;
    else { PRINT_rev ( P.getNEXT() );
           System.out.PRINTLN (P.getDATA()) ;
         }
}
 void PRINT_reversed() {
      PRINT_rev(head);
 }
```

Key 4

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 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 int compares;

   public SimpleTree() {
      root = null;
      compares = 0;
   }
   public int getCompares() {
      return compares;
   }
   public void clearCompares() {
      compares = 0;
   }

   public void add(int x){

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

key 5

```
            q = t;
            t = t.rc;
          }
        } // end while
        compares++;
        if(x < q.data) {
          q.lc = new Node(null, x, null, q);
        }
        else {
          q.rc = new Node(null, x, null, q);
        }
      }
    } // end add
    public void traversal(Node r) {
      if(r == null) return;
      if(r.rc != null)traversal(r.rc);
      System.out.println(r.data);
      if(r.lc != null)traversal(r.lc);
    }
    public void traversal() {
      traversal(root);
    }
    public static void main(String[] args) {
      SimpleTree st = new SimpleTree();
      // Question 2 a has five lines of output
      st.add(6);
      System.out.println("C = " + st.getCompares()); // Line 1
      st.clearCompares();
      st.add(2);
      System.out.println("C = " + st.getCompares()); // Line 2
      st.clearCompares();
      st.add(8);
      System.out.println("C = " + st.getCompares()); // Line 3
      st.clearCompares();
      st.add(15);
      System.out.println("C = " + st.getCompares()); // Line 4
      st.clearCompares();
      st.add(18);
      System.out.println("C = " + st.getCompares());   Line 5

      Question 2 b
      st.traversal();
    }
}
```

Key

2.(a) What are the first five lines of output? (5 Points)

1
3
3
7
5

2.(b) What will the program display when the traversal method is called. ? (5 Points)

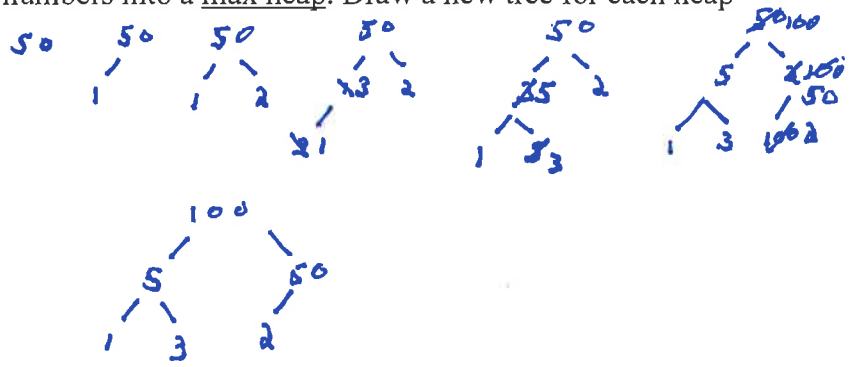
18
16
8
6
2

key 7

Heaps (12 points)

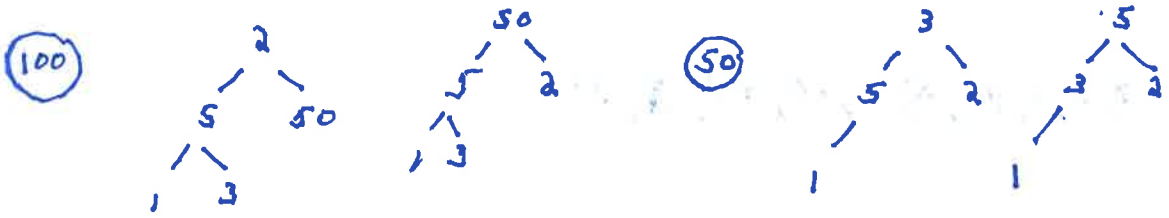
3) Insert the following 6 numbers into a max heap. Draw a new tree for each heap insertion. (4 Points)

50, 1, 2, 3, 5, 100



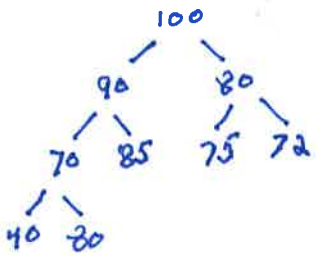
4) What is the height of the tree that you drew in question 3? (A single node in a tree gives a height of 0.)
(2 Points) 2

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 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.

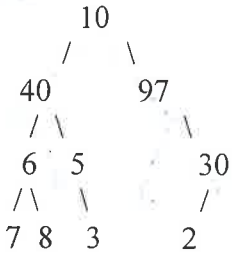
100
90
80
70 ✓
85
75
72
40
80 ✓



key 8

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, 7, 8, 5, 3, 97, 30, 2

(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)

7, 6, 8, 40, 5, 3, 10, 97, 2, 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)

10, 40, 97, 6, 5, 30, 7, 8, 3, 2

(d) 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 leaves will the tree have? (2 points) 2^h Note, this tree has a perfectly flat bottom. We need the total number of leaves in terms of h. This is an exact answer, not Big O.

0 $2^0 = 1$

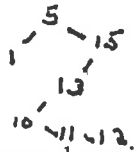
$2^1 = 2$

$2^2 = 4$

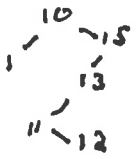
(e) In general, if a binary tree (at most two children per node) is perfectly balanced (unlike the tree pictured above) and complete with height h, how many internal nodes (non-leaves) will the tree have? (2 points) $2^h - 1$
Note, this tree has a perfectly flat bottom. We need the total number of internal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

KaY 9

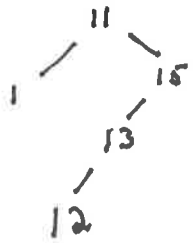
8. (a) Insert he following numbers into a Binary Search Tree. Draw the tree after all insertions are complete. (1 Point)
5, 1, 15, 13, 10, 11, 12



(b) Delete 5 from the final tree that you drew in 8 (a). We are following the "go right once, left hard" rule. Draw this final tree. (1 Point)



(c) Delete 10 from the final tree in question 8 (b). Again, we are following the "go right once, left hard" rule. Draw this final tree. (3 Points)



Project Questions (20 points)

Key 10

(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, 1, 7, 101} and **k** = 14. 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. a decision problem.
5. an optimization problem.

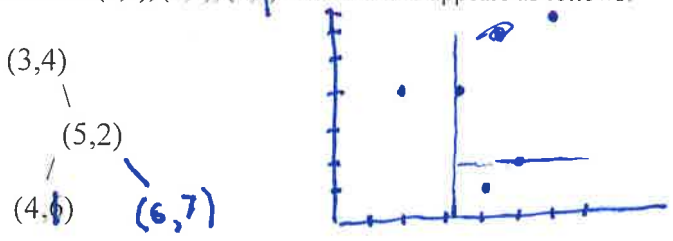
(c) Suppose Alice sends the integer K to Bob. K is computed using Bob's Merkle-Hellman public key combined with the message M. The central idea behind Merkle-Hellman is that a potential eavesdropper could read the message M if the eavesdropper could (circle the one best option) (1 Points)

1. Find K so that M is prime.
2. Modify Bob's public key.
3. Find a subset of Bob's public key that sums to K.
4. Modify the super increasing sequence.
5. Find a subset of a super increasing sequence 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 7 mod 13? 2 (2 Points)

$2 \cdot 7 \equiv 14 \bmod 13 \equiv 1$

(e) The following points, in a standard (x,y) coordinate plane, have been added to a 2-d tree. (3,4), (5,2), (4,6). The 2-d tree appears as follows:



Add the point (6,7) 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 (6,7), in (e). Suppose that we performed a nearest neighbor search for the point (2,4). Which points in the tree need to be examined? (3,4)
(2 Points)

Key 11

(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)

MAX OFF == 4

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

(h)The following is a data file for Project 3. Note the course Hist4 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 Hist4 in the Red-Black tree of Project 3? (4 Points) 3

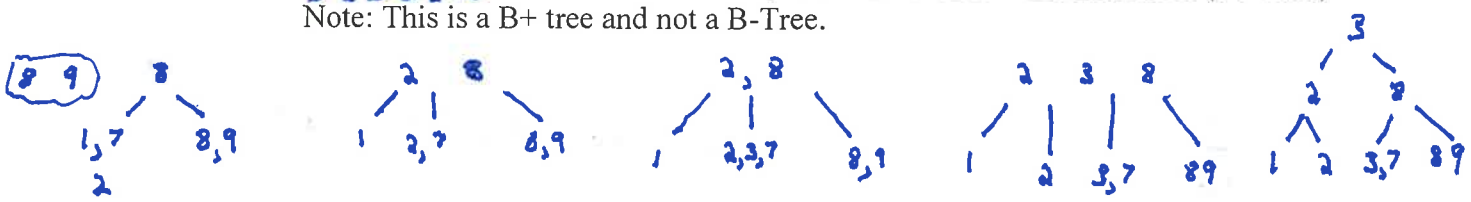
0 1 2
Amy Calc1 Span1 Philo2
Bill Calc1 Philo2 Hist4 Logic2
0 2 3 4

(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 "Hist4" have? (3 Points) 3

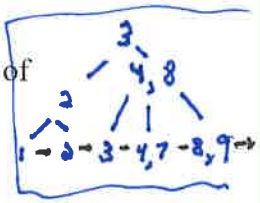
B and B+ Trees (21 points)

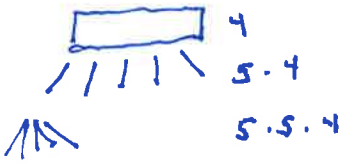
10. (a) Insert the following numbers into a B-Tree with a minimum of 1.
9, 8, 7, 1, 2, 3, 4 Draw the final tree. (7 Points)

(b) Insert the following numbers into a B+ Tree with a minimum of 1.
9, 8, 7, 1, 2, 3, 4 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 2. What is the exact maximum number of keys such a tree could hold if the tree were of height 2? _____
(7 Points)





Key, 2

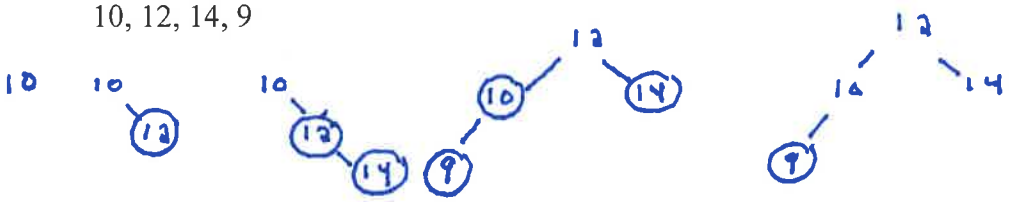
Red Black Trees (8 points)

Key 13

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, 12, 14, 9



Graph Representations (8 points)

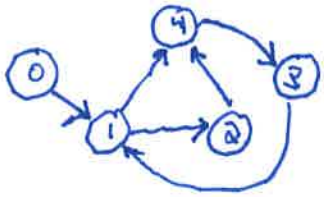
Consider a directed graph G_1. The graph is represented by an adjacency matrix m. If there is an edge from i to j then m[i,j] = true.

Matrix m

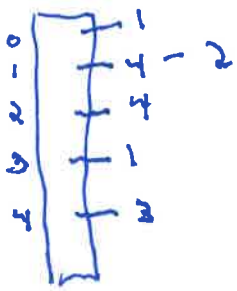
vertex	0	1	2	3	4	5
0		true				
1			true		true	
2					true	
3		true				
4				true		

G_1

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



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. (6 Points)



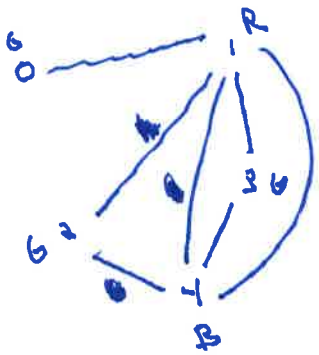
Key

Graph Coloring (10 points)

13. Color the graph G_2 with as few colors as possible. Use convenient colors - such as red, blue, 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	5
0		true				
1	true		true	true	true	
2		true			true	
3		true			true	
4		true	true	true		

G_2



Key, 4

Big O (8 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^2).
(b.) In the best case, the algorithm is Big O(2^n).
(c.) In the best case, the algorithm is Big O(n!).
d. In the worst case, the algorithm is Big Theta(n^2).
(e.) In the best case, the algorithm is Big Theta(n).
(f.) In the worst case, the algorithm is Big Omega(n).

b
c
f

15. 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 have a new IoT gadget in my pocket. I click on the gadget and my car begins to beep loudly. I hear my car and walk directly to it. This algorithm (circle all that are correct). (4 Points)

(a.) runs in Big O(n!).
(b.) runs in big O(n).
(c.) runs in Big Theta(1).
(d.) runs in Big Omega(1).
e. runs in Big Omega(n).
f. runs in Big Omega(n!).
(g.) runs in Big O(n^3).
(h.) runs in Big O(1).

K.4,5