95-771 Data-Structures and Algorithms
Fall 2013

Midterm Exam
Carnegie Mellon University

Print Full Name__ Key ____________________

There are 110 points on this exam. Your score will be computed as a percentage of 110.

Coding Lists and Trees (18 points)

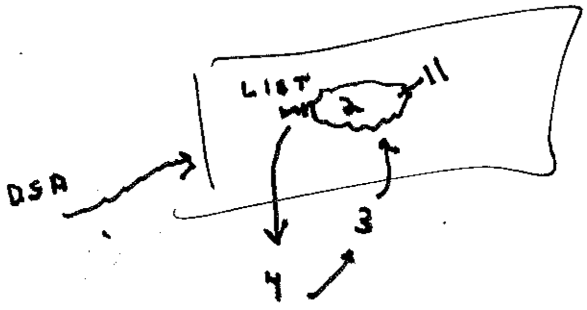
1. The following program contains four println() statements. Describe the output of the program when each of the println() statements is executed. (8 points):

```
package dsamidtermproject;

class Node {
  private int data;
  private Node next;

  public Node(int data, Node next) {
     this.data = data;
     this.next = 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 class DSAMidtermProject {
  public Node list = null;

  public void insert1(int x) {

     list = new Node(x,list);
  }

  public void insert2(int x) {
     if(list == null) insert1(x);
     else {
        Node m = list;
        Node p = list;
        while (m != null) {
           p = m;
           m = m.getNext();
        }
        p.setNext(new Node(x,null));
     }
  }
```

6, 10c

Key

```
  public void adjust() {

    if(list == null) return;
    else {
      Node m = list;
      Node p = list;
      while (m != null) {
        p = m;
        m = m.getNext();
      }
      p.setNext(list);
    }
  }
  public String toString() {
    String v = "";
    Node c = list;
    while (c != null) {
      v = v + c.getData();
      if(c.getNext()!= null) v = v + "->";
      c = c.getNext();
    }
    return v;
  }
```



```
  public static void main(String args[]) {

    DSAMidtermProject dsa = new DSAMidtermProject();

    dsa.insert2(2);
    dsa.insert1(3);
    dsa.insert1(4);

    // Question 1.a
    System.out.println(dsa);
```

4 -> 3 -> 2

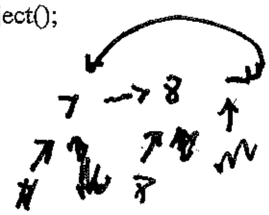
LIST -> 5 -> 6 -||

```
    dsa = new DSAMidtermProject();
    dsa.insert2(5);
    dsa.insert2(6);

    // Question 1.b
    System.out.println(dsa);
```

5 -> 6

NO ARROWS
-1

```
    dsa = new DSAMidtermProject();
    dsa.insert1(7);
    dsa.insert2(8);

    // Question 1.c
    System.out.println(dsa);
```



```
    dsa.adjust();
    // Question 1.d
    System.out.println(dsa);
  }
}
```

LOOPS FOREVER

DID NOT handle
new LIST
-2

-1 IF ANY LIST IS LONG

KEY

1.a What will the program display at the println marked 1.a?

4 -> 3 -> 2

1.b What will the program display at the println marked 1.b?

5 -> 6

1.c What will the program display at the println marked 1.c?

7 -> 8

1.d What will the program display at the println marked 1.d?

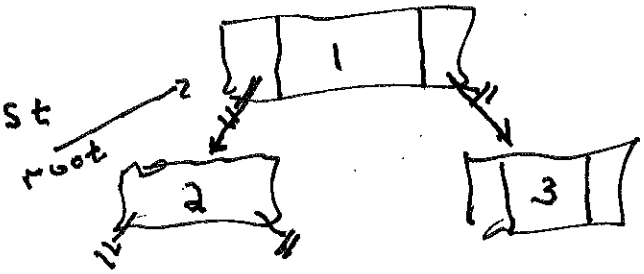
LOOPS No display

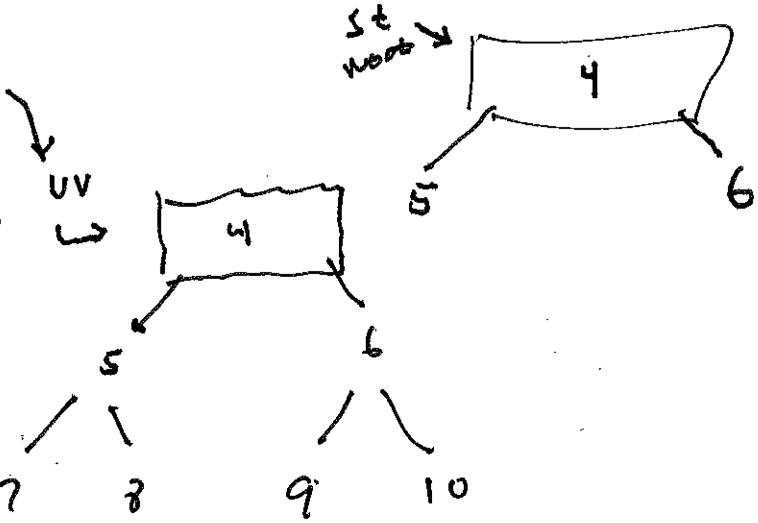
2. The following program contains one System.out.print() statement. (10 points):

```
package simpletree;
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 static int ctr = 0;

   public SimpleTree() {
      root = null;
   }

   private Node add(Node t){

      if (t == null) {
         ctr = ctr + 1;
         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 + 1;
         root = new Node(null,ctr,null);
      }
      else {
         add(root);
      }
   }
```

Key

```
public void traversal(Node t) {

   if(t != null) {
      System.out.print(t.data + ":");
      traversal(t.lc);
      traversal(t.rc);
   }
}
public void traversal() {
   traversal(root);
}

private Node xyz(Node t)
{
  if (t == null) return null;
  else
  {
     Node lc =
         xyz(t.lc);
     Node rc =
         xyz(t.rc);
     return new Node(lc,t.data,rc);
  }
}
public SimpleTree xyz() {
   if (root == null) {
      return new SimpleTree();
   }
   else {
      SimpleTree newSt = new SimpleTree();
      newSt.root = xyz(this.root);
      return newSt;
   }
}
public static void main(String[] args) {
   SimpleTree st = new SimpleTree();
   st.add();
   st.add();
   // Question 2.a      1: 2: 3:
   st.traversal();

   st = new SimpleTree();
   st.add();
   st.add();
   SimpleTree uv = st.xyz();
   uv.add();
   // Question 2.b
   uv.traversal();
  }   4:5:6:7:8:9:10
}
```





Key

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

1 : 2 : 3

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

4:5:7:8:6:9:10

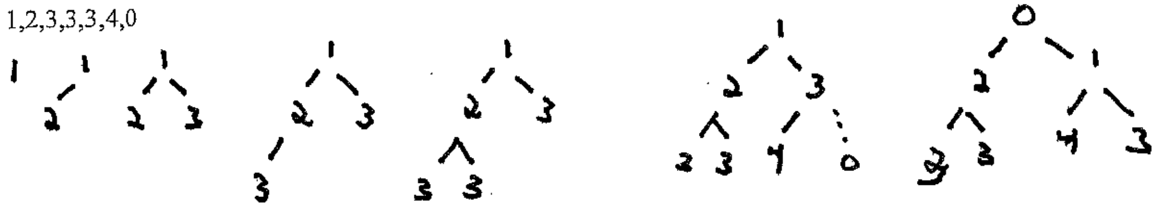
-1 IF Provided A Fresh START At 1:2··

1,2,4,5,3,6,7

Heaps (12 points)

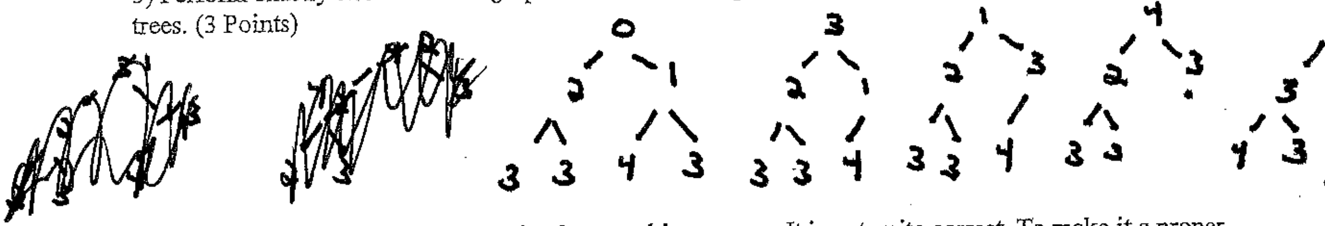
3) Insert the following 7 numbers into a min heap. Draw a new tree for each heap insertion. (4 Points)

1,2,3,3,3,4,0



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

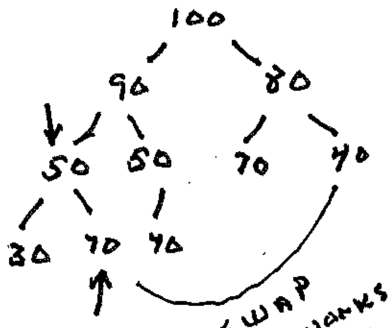
5) Perform exactly two deleteMin() operations on the heap that you drew in question 3. Draw the resulting trees. (3 Points)



4 could be either child

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 need to be swapped in order to make this a max heap? (3 points)

100
90
80
50 ✓
50
70 ✓
40
30
70 ✓
40



SWAP here works but is NOT IN the SPIRIT OF a heAP, 50, -1

Key

Binary Trees (16 points)

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

```
              20
           /      \
         40        97
        / \        / \
       6   9      2   76
      / \   \    /    / \
     5   1   8  3   65    7
```

(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, 1, 9, 8, ~~20~~, 97, 2, 3, 76, 65, 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)

5, 6, 1, 40, 9, 8, 20, 3, 2, 97, 65, 76, 7

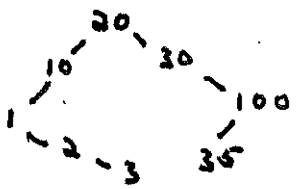
(c) List the data that would be accessed by a level-order traversal on the given tree by writing out the values in the nodes as they would be accessed, separated by commas. (2 points)

20, 40, 97, 6, 9, 2, 76, 5, 1, 8, 3, 65, 7

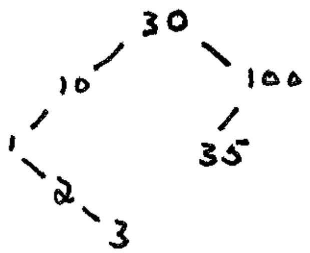
(d) In general, if a binary tree is perfectly balanced (unlike the tree pictured here) and complete with height h, how many leaves, in terms of h, will the tree have? (1 point) 2^h Note, this tree has a perfectly flat bottom. In addition, how many internal nodes would such a tree have (in terms of h)? (1 Point) $2^h - 1$

(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) $\text{Log}_2 K$ 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)
20, 10, 30, 1, 2, 3, 100, 35



(b) Delete 20 from the final tree that you drew in 8 (a). Draw this final tree. (2 Points)



95-771 Data Structures and Algorithms
Fall 2013

Midterm Exam
Carnegie Mellon University

Key /

Project Questions (18 points)

9. Recall the Merkle-Hellman cryptosystem that we worked with in Project 1, the spell checker application in Project 2, and the calculator problem from Project 3.

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 = {3,9,12,4} and **k** = 17. Is there a subset of X which sums to **k**?
NO ____ 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.

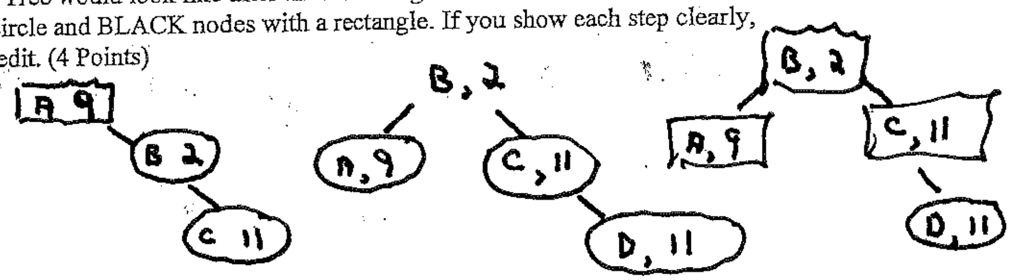
(c) Suppose Alice sends a message (K) to Bob. K is computed using Bob's Merkle-Hellman public key combined with the message M. A central idea behind Merkle-Hellman is that a potential eaves dropper could read the message M if the eaves dropper could (2 Points) Circle the one best answer.

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}) mod m=1. What is the modular inverse of 3 mod 5? __2__ (4 Points)

(e) In Project 3, we wrote a calculator that processed RPN expressions and used a Red Black Tree. Draw what the Red Black Tree would look like after the following user interaction. Draw RED nodes with a circle and BLACK nodes with a rectangle. If you show each step clearly, you will receive partial credit. (4 Points)

A 4 =
A 9 =
B 2 =
C A 2 + =
D C=

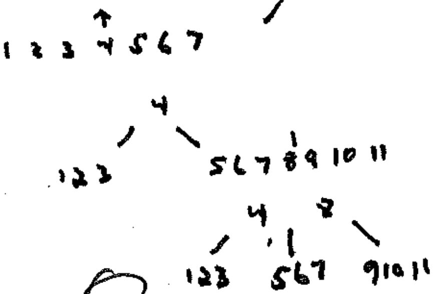


(f) In Project 2 we wrote a spell checker that loaded n words into a Red Black tree and allowed a user to make queries against the tree. We wrote a lookup method that checked if a word was present. Which of the following is true of the worse-case lookup method? Circle all of those that are true. (You may or may not have more than one answer.) (4 Points)

[✓ 1.] It ran in O(LogN)
2. It ran in O(1)
3. It ran in $\Omega(N^2)$
[✓ 4.] It ran in O(N)
[✓ 5.] It ran in Θ(LogN)
[✓ 6.] It ran in $O(2^n)$

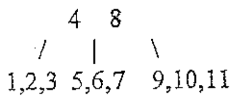
-1 for each error maximum of 4 off

Key



Balanced Trees (21 points)

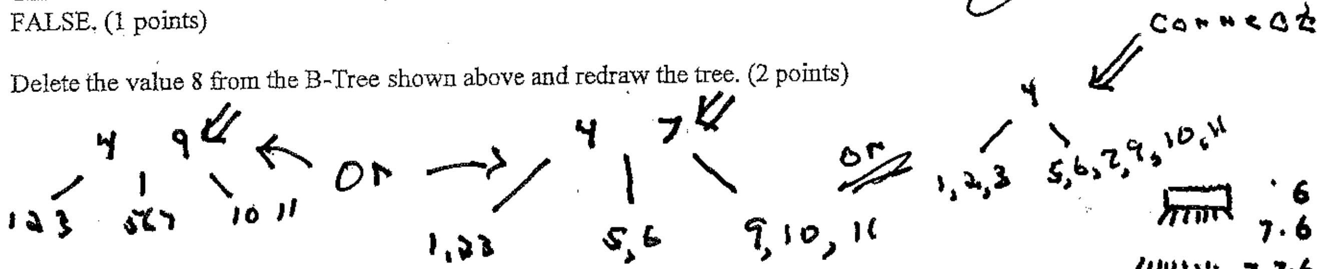
10. Consider the following B-Tree with a minimum of three.



(a) This tree could have been created by inserting 1,2,3,4,5,6,7,8,9,10,11 (in this order). Circle [✓ TRUE] or FALSE. (1 points)

(b) This tree could have been created by inserting 11,10,9,8,7,6,5,4,3,2,1 (in this order). Circle [✓ TRUE] or FALSE. (1 points)

(c) Delete the value 8 from the B-Tree shown above and redraw the tree. (2 points)



.6
7.6
7.7.6

(d) Consider again the original tree of height of 1. What is the maximum number of keys that this type of tree (min = 3) could hold with a height of 2? (1 point). 342

6
+42
294
342

(e) Consider again the original tree of height of 1. Write a formula that would allow one to calculate the maximum number of keys that could be held by such a tree with height h. (2 points).

$$\cancel{\sum_{i=0}^{h} 7^i \cdot 6} = 7^0 \cdot 6 + 7^1 \cdot 6 + 7^2 \cdot 6 + \ldots + 7^h \cdot 6$$

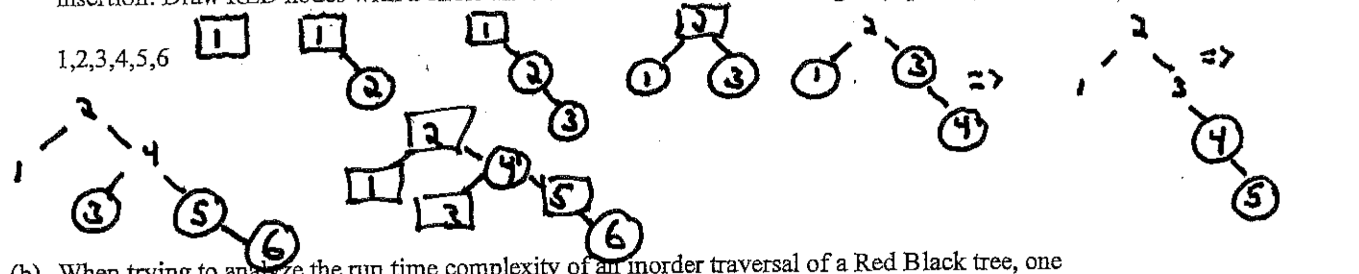
$$= 6 \cdot \sum_{i=0}^{h} 7^i \quad \text{or} \quad 7^{h+1} - 1$$

42
·7
294

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 and BLACK nodes with a rectangle. (5 points)

1,2,3,4,5,6



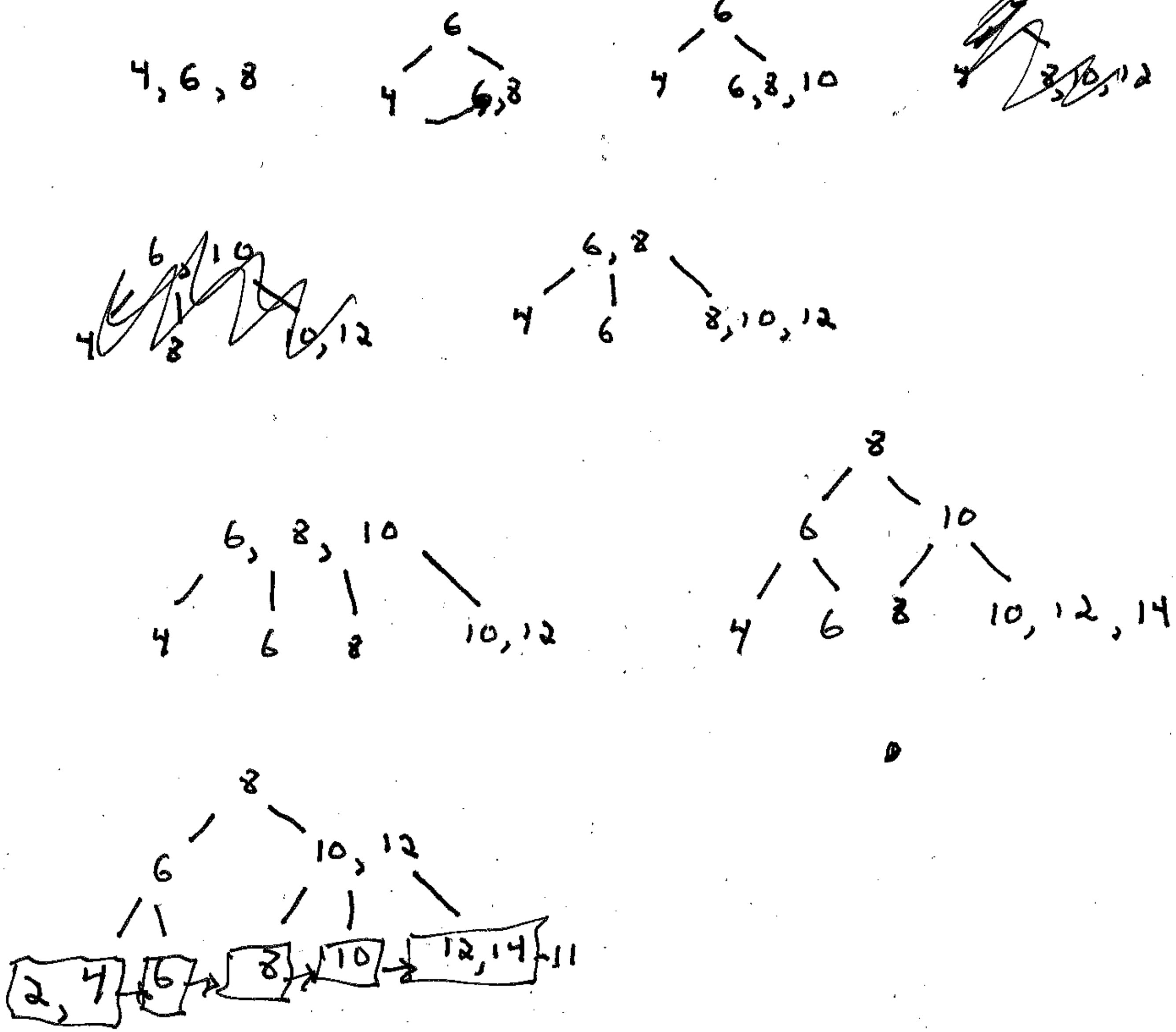
(b) When trying to analyze the run time complexity of an inorder traversal of a Red Black tree, one should consider the worst case, average case and best case separately. Each case may have a different run time performance. Circle TRUE or [✓ FALSE] (1 point)

(c) What is the worst-case runtime complexity of a Red Black Tree lookup operation? Use Big Theta notation. (2 points) $\Theta(\log_2 N)$

KeY

12. B+ Trees. Enter the following numbers into a B+ Tree with minimum = 1 and maximum = 2. Draw the tree after each insertion. The integer 4 is added first, six is added second and so on. (6 Points)

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



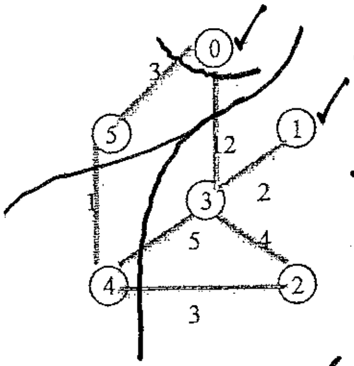
Key

Graph Algorithms(25 points)

13. (a) Consider the weighted and undirected graph below. When starting at vertex 0, breadth first search (BFS) would find what path from vertex 0 to vertex 1. The path should be expressed as a list of vertices. (3 points) 0, 3, 1

(b) Consider the weighted and undirected graph below. When starting at vertex 0, and by first selecting vertex 5, show one path that depth first search (DFS) might find. (3 points) 0, 5, 4, 3, 1 Other paths Are Possible

(c) Draw the contents of the distance array for each iteration of Prim's Algorithm as it works on this graph. 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. (8 points)



	0	1	2	3	4	5
0	0	?	?	?	?	?
5	0//			12		3//
4	0//			12	1	3//

-6 IF NOT connect order

	0	1	2	3	4	5
2	0//		3	5	1//	3//
3	0//		3//	4	1//	3//
1	0	2	3	4	1	3

-2 IF NOT connect

(d) Draw an adjacency matrix representation of the graph shown immediately above. In this adjacency matrix representation, -1 is used to denote that no edge exists between two vertices. (2 points)

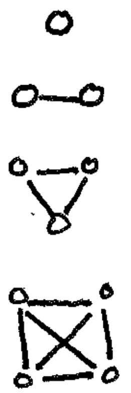
	0	1	2	3	4	5
0				12		3
1				2		
2				4	3	
3						
4				5		
5				etc.		

(e) The graph shown above is a simple, undirected graph. It contains no loops or multiple edges. Suppose we have such a graph with V vertices. What is the maximum number of edges that such a graph can contain? Your answer should be an exact formula. Answers in Big Theta notation don't count. (1 point)

$V(V-1)/2$

or

$$\frac{V^2 - V}{2}$$



V	Edges
1	0
2	1
3	3
4	6
⋮	$v \cdot (v-1)/2$

must be exact.

95-771 Data Structures and Algorithms
Fall 2013

Midterm Exam
Carnegie Mellon University

Key

(f) Using the Floyd Warshall algorithm shown below, draw a matrix each time the comment "draw cost matrix" is encountered in the code. Use the graph below for input. You may assume that the original graph is represented in an adjacency matrix c. Also, you must assume that the loops proceed through the graph in numerical order. That is, the first vertex is vertex 0 and the second is vertex 1 and so on. (8 points)

```
for each vertex u in G do {
   for each vertex v in G do {
      cost[u,v] = c[u,v]
   }
}

// Now, draw the cost matrix in the first box

for each vertex w in G do {
   for each vertex u in G do {
      for each vertex v in G do {
         cost[u,v] = min(cost[u,v],cost[u,w] + cost[w,v]
      } // end for v
   } // end for u
   // Now, draw the cost matrix in the next box
} // end for w
```

Cost

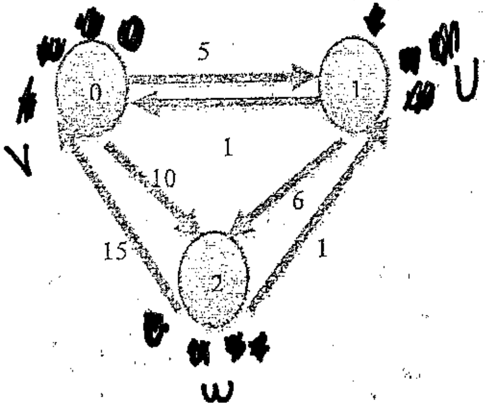
	0	1	2
0		5	10
1	1		6
2	15	1	

	0	1	2
0		5	10
1	1		6
2	15	1	

	0	1	2
0		5	10
1	1		6
2	2	1	

change 2 TO here.

	0	1	2
0		5	10
1	1		6
2	2	1	


0
5
1
U
1
V
10
6
15
1
2
W