CMPS 212 Final Exam
American University of Beirut
Department of Computer Science

Date: January 29th, 2004

This is a closed book exam
You have two hours to finish
Answer in the spaces provided. Use additional sheet when needed.
No electronic devices are allowed

The questions are not all of equal difficulty, do the easy ones first.

Good Luck

Name :

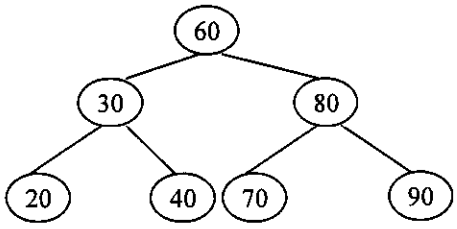
Student # :

Section :

Part I: Multiple Choices	**/29**
Part III: Concept questions	**/51**
Part IV: Programming	**/20**

Multiple choices [29 points]

1. Given the following binary search tree, A [4 points]:



Which of the following sequences represents a postorder traversal of a binary search tree, B which is obtained by first removing 60 from A, and then inserting 60 back into the resulting tree:

a) 20 30 40 60 70 80 90

b) 70 30 40 60 20 80 90

c) 20 60 40 30 90 80 70

d) 20 40 60 30 80 90 70

e) 70 30 20 40 60 80 90

f) none of the above

2. A complete tree of height h has [3 points]

a. 2^h nodes

b. 2^{h+1}-1 nodes

c. between 2^h and 2^{h+1}-1 nodes

d. 2^{h-1} nodes

e. between 2^{h-1} and 2^h+1 nodes tree of height h has

f. None of the above

3. Which of the following is not a red-black rule [3 points]?

a. Every path from a root to a leaf, or to a null child, must contain the same number of black nodes.

b. If a node is black, its children must be red

c. The root is always black

d. None of the above.

4. **Which of the tables of size 7 below represents a hash table with [4 points]:**
   - quadratic probe conflict resolution
   - hashing function h(k) = k mod 7
   after inserting : 16, 17, 9, 23, 24, 03

a)

0	1	2	3	4	5	6
24	03	16	17	23		09

b)

0	1	2	3	4	5	6
24		16	17	23	03	09

c)

0	1	2	3	4	5	6
	24	16	17	23	03	09

d)

0	1	2	3	4	5	6
03		16	17	23	24	09

e)

0	1	2	3	4	5	6
24	23	16	17	03		09

f) none of the above

5. **Which of the following arrays does NOT represent a heap [3 points].**

   a) 200 113 150 102 113 149 001

   b) 200 113 150 102 112 149 161

   c) 200 113 150 102 112 149 001

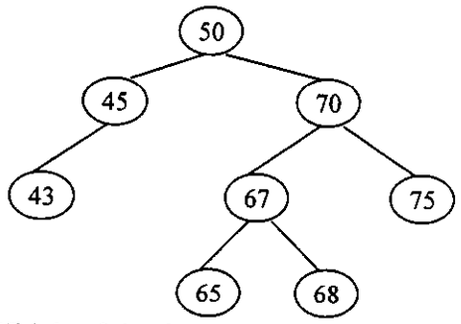
   d) 200 113 150 102 112 001 002

   e) 201 114 151 103 113 150 004

   f) none of the above

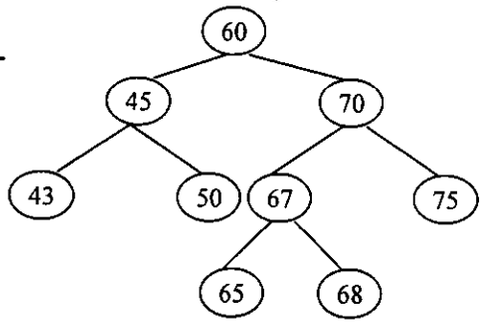
6. **How many leaves does a *full binary tree* of *n* nodes have? [3 points]**
   a. n/2
   b. (n+1)/2
   c. (n-1)/2
   d. none of the above

7. **Given AVL tree, A below [3 points]**

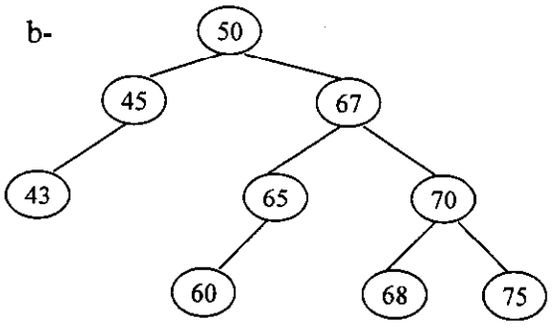


Which of the following trees is the result of inserting 60 into A:

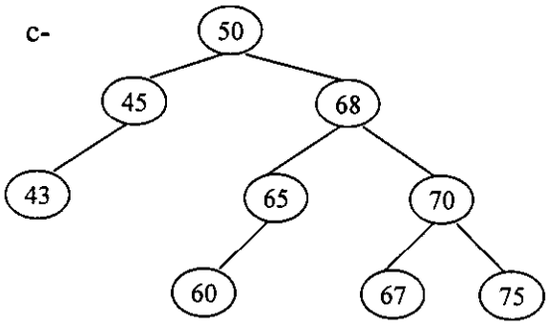
a-



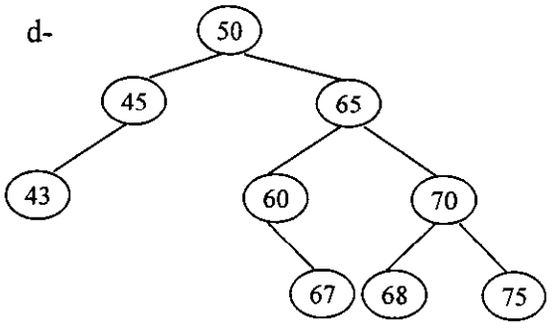
b-



c-



d-



e- none of the above

8. **Consider the graph represented by the following adjacency list [3 points]:**

1: 2—3—6
2: 5—1—3
3: 2—1—7—5—6
4: 5—7
5: 2—4—3—6
6: 3—5—1
7: 3—4

Perform a Breadth First Search in the graph starting from node 1 and processing the edges adjacent to a node in the order they appear in the adjacency list.
What is the order in which the nodes are visited?

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

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

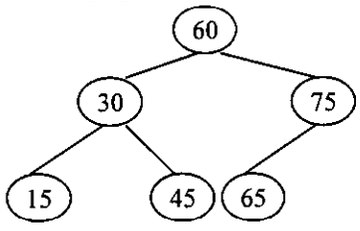
c) 1,2,3,6,7,5,4

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

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

f) none of the above

9. **Below are several traversal algorithms for binary trees. Which one would give the keys in decreasing order [3 points]?**



a) visit left sub-tree
visit the root
visit right sub-tree

b) visit right sub-tree
visit the root
visit left sub-tree

c) visit the root
visit left sub-tree
visit right sub-tree

d) visit right sub-tree
visit left sub-tree
visit the root

e) None of the above

Concept Questions [51 points]

Question I. Hash Table [8 points]

1. **Show the results of storing the following data in the order in which it is listed in the hash table below using k mod 11 as the hash function and linear probing as the collision resolution technique. Data: 4, 18, 20, 22, 15, 16, 17, 5 [4 points]**

0	1	2	3	4	5	6	7	8	9	10

2. **Repeat the above using double hashing. The secondary hash function is (7 - k mod 7). [4 points]**

0	1	2	3	4	5	6	7	8	9	10

Question II: Huffman Code [13 points]

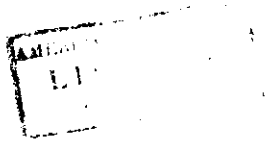
1. **Draw the Huffman tree for the following sentence [8 points]:**
   a man a plan a canal panama

Frequencies

a	
c	
l	
m	
n	
p	
sp	

sp stands for space

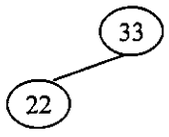
2. **How would the word "man" be encoded using your Huffman code? [3 points]**



3. **Why is it always possible to reconstruct a message that has been encoded using a Huffman code? [3 points]**

Question IV: AVL and Red Black tree [8 points]

1. Consider that you have the following AVL tree, T

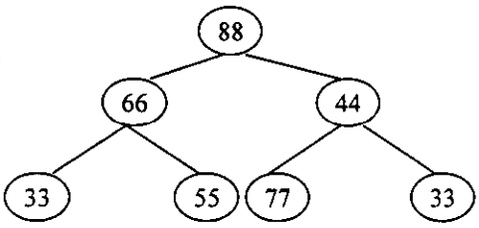


Insert the following nodes into T: 11 then 55 then 44 [5 points]

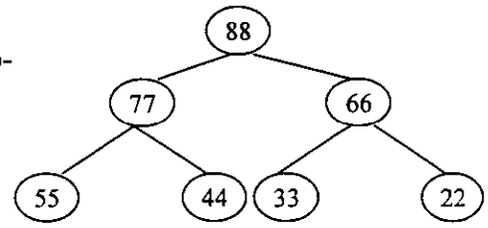
Question III: Heap and heap sort [8 points]

1. Determine which of the following binary trees is a heap [3 points]

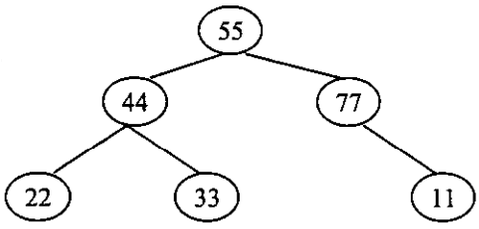
a-



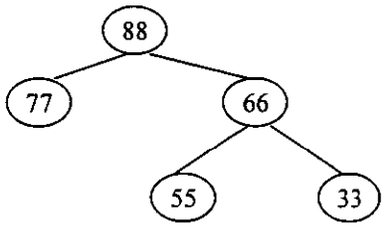
b-



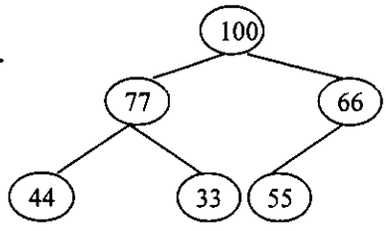
c-



d-



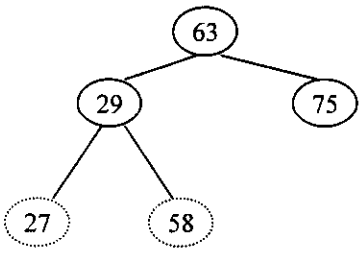
e-



f-

2. Explain why a heap-sort has complexity O(nlogn) [5 points]

2. Insert node with key 20 into the following red black tree (◯ represents red node) [3 points]

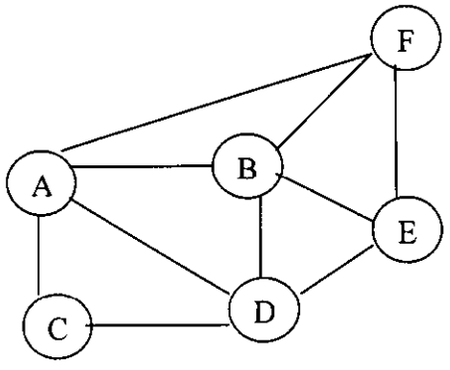


Question V: Graph [13 points]

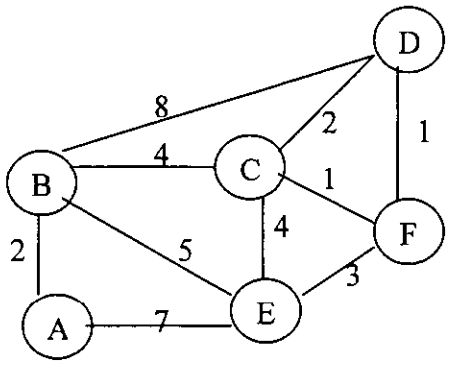
1. **Given the below adjacency matrix of a graph, draw the correspondent adjacency list.[2 points]**

	A	B	C	D
A	0	1	1	0
B	0	1	1	1
C	0	0	0	0
D	1	1	1	0

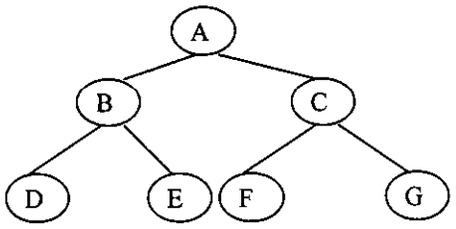
2. **Starting with vertex A, edge AB, show the spanning tree which results from a breadth first search of the undirected graph below assuming that, in case of several choices, vertices are selected in alphabetical order [3 points].**



3. **Show the minimum spanning tree which results from applying Kruskals algorithm on the weighted graph below. [4 points]**



4. **If the tree below is regarded as a graph**



4.1 Give the output of using the depth first search algorithm to traverse the tree starting with vertex A. When there are several choices, vertices are selected in alphabetical order. (output is sequence of nodes visited first) [2 points]

4.2 Output from depth-first tree traversal is similar to output from [2 points]:

a) inorder traversal
b) postorder traversal
c) preorder traversal
d) none of the above

Programming Questions [20 points]

1. **Suppose that the classes Node and Tree are used to implement a binary tree data structure. Complete the method findSibling (or write one on your own) in class tree that takes one parameter key. The method must return the sibling of the node key if found. Otherwise, if the key node or the sibling node is not found in the tree, the method returns null.**

```
class Node {
        public int iData;
        public Node leftChild;
        public Node rightChild;
}
class Tree
{
          private Node root;
          public Node findSibling (int key) { … }
          …
}

public Node findsibling(int key)
{
        Node current = ……..

        Node sibling=……;

        while(current.iData != key)
        {
                if(……..)
                {
                        ……..

                        ……
                }
                else
                {
                        ……..

                        ……..
                }
                if(current == null)

                        ……

        }

        ……..
}
```

2. **Suppose that the classes DataItem and HashTable are used to implement a hash table data structure. Complete (or write one on your own) the method findKey() in class HashTable that takes one parameter key. It returns true if the key is in this table (using quadratic probing), and it returns false otherwise**

```
class DataItem {
        public int iData;
        public DataItem(int data)
        {
                iData = data;
        }
} // end class DataItem

class HashTable {
        DataItem[] hashArray;                   // array holds hash table
        int arraySize;
        DataItem nonItem;                       // for deleted items
        public HashTable(int size) {            // constructor
                arraySize = size;
                hashArray = new DataItem[arraySize];
                nonItem = new DataItem(-1);          // deleted item key is -1
        }
        public boolean findKey(int key){...}
}

public boolean findKey(int key)
{
        int hashVal = ……..
         int index = hashVal;
        int i=1;
        int count = 0;
        while(…….. &&  count < ……)
        {
                if (……)

                        ……..

                index = ……

                ……

                ……
        }
        ……
}
```