CSCI 2110
DATA STRUCTURES AND ALGORITHMS

TOPICS AND PRACTICE QUESTION BANK FOR FINAL EXAM

NOTE:
FINAL EXAM IS ON THURSDAY, DECEMBER 15th, FROM 3.30 PM
DURATION: 2 HOURS 30 MINUTES
LOCATION: McCAIN ARTS AND SOCIAL SCIENCES BUILDING AUD-1 and AUD-2
FORMAT: CLOSED BOOK, PROCTORED EXAM ON BRIGHTSPACE WITH RESPONDUS LOCKDOWN BROWSER

DETAILED BREAKDOWN OF TOPICS

Module 4: ORDERED LISTS
- Definition and examples
- BINARY SEARCH algorithm and its complexity
- Generic Ordered List Class – Know how the methods work and how to use them
- Merging two ordered lists – the two-finger walking algorithm

Module 5: RECURSION
- What is a recursive definition?
- Recursion: Definition, Advantages, Base and Glue Cases
- Examples: Writing recursive methods
- Examples: Tracing recursive methods

Module 6: BINARY TREES
- Definition and Terminology
- Strictly binary and complete binary trees
- Traversals: Preorder, Inorder, Postorder, Level Order
- Recursive definitions for a binary tree
- Binary Tree Class: Methods and Complexities
- Application: Huffman coding

Module 7: BINARY SEARCH TREES
- Definition and Examples
- Operations – Search, Insert and Delete
- Binary Search Tree Class: Methods and Complexities

Module 8: HASHING AND HASH TABLES
- Motivation
- Hashing, Hash Function and Hash Table Concept with Examples
- Hash Clash
- Solution to Hash Clash: Open Hashing or Separate Chaining
- Solution to Hash Clash: Closed Hashing – Linear Probing and Quadratic Probing
- Miscellaneous Concepts in Hashing
- HashMap in Java with examples

Module 9: SORTING ALGORITHMS

- Motivation and applications of sorting
- Bubble Sort: Algorithm, Examples and Complexity
- Selection Sort: Algorithm, Examples and Complexity
- Merge Sort: Algorithm, Examples and Complexity
- Bucket Sort: Algorithm, Examples and Complexity

Module 10: GRAPHS

- Definition and Motivation
- Graph Terminology with Examples
- Graph Representation: Adjacency Matrix and Adjacency List
- Graph Algorithms
- Graph Traversals: Depth First and Breadth First
- Topological Sorting
- Shortest Path Algorithms

NOTES:

1. For the OrderedList class, BinaryTree class, and BinarySearchTree class, you should know how the methods work. For programming questions pertaining to these classes, you will be given the list of methods from the class and their meanings. You may be asked to develop other methods using these methods. You may be given code snippets and asked to fill in the missing lines of code. You may also be given code snippets and asked to trace the output.

2. You will not be asked to draw any diagrams on the Brightspace exam portal. Instead, you may have to draw diagrams on a scratch sheet of paper, and then enter the answers in text on the Brightspace exam portal.

CSCI 2110 DATA STRUCTURES AND ALGORITHMS
PRACTICE QUESTION BANK FOR THE FINAL EXAM

Question 1. Multiple Choice Set - For each of the following questions, select the most appropriate answer:

1. Which of the following statements is correct about the Ordered List data structure?
   I. It is a linear data structure.
   II. The items are arranged in ascending or descending order of keys.
   III. Repetition of keys is allowed.

   a. I only
   b. I and II
   c. II and III
   d. I, II and III

2. What is the complexity of merging two lists with n items each using the two-finger walking algorithm?
   a. $O(n^2)$
   b. $O(\log_2 n)$
   c. $O(n\log_2 n)$
   d. $O(n)$

3. The maximum number of searches required by the binary search algorithm to search an ordered list of n items, where n is a power of 2, is
   a. n
   b. n+1
   c. $\log_2 n$
   d. $\log_2 n + 1$

4. Suppose that a recursive method implements the recursive definition given below:
```
square (1) = 1
square (N) = square (N-2) + 2*N – 1
```
   Assuming that the definition has been implemented correctly, how many times will the method be called if the main method calls square(5)?
   a. 5
   b. 3
   c. 6
   d. 1

   square(5)
   square(3) → square(1)

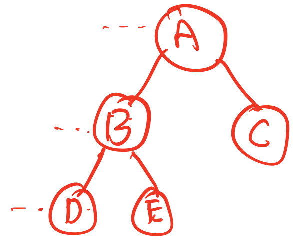
5. The number of branches from the root to a node in a binary tree is called
   a. height of the node
   b. depth of the node
   c. height of the root
   d. depth of the root

6. A binary tree, in which all levels are filled, except possibly the last level, which is filled from left to right, is called a
   a. Binary Search Tree
   b. Complete Binary Tree
   c. Strictly Binary Tree
   d. Leveled Tree

7. Which of the following statements about binary trees is NOT true?

  a. Every binary tree has at least one node.
  b. Every non-empty tree has exactly one root node.
  c. Every node has at most two children.
  d. Every non-root node has exactly one parent.

The next three questions pertain to a binary tree with 5 nodes. The root is A. A has a left child B and right child C. B has a left child D and a right child E. There are no other nodes.



Pre (Root-L-R):
ABDEC
In (L-Root-R)
DBEAC
Post (L-R-Root)
DEBCA
Level order
ABCDE

8. Which of the following traversals yields ABCDE?

  a. Inorder
  b. Level order
  c. Post order
  d. Pre order
  e. Two of the above

9. Which of the following is an inorder traversal of the tree?

  a. ABCDE
  b. ABDEC
  c. DBEAC
  d. DEBAC
  e. None of the above

10. The height of the tree is

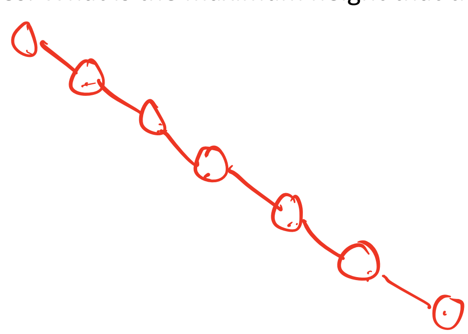
  a. 0
  b. 1
  c. 2
  d. 3
  e. None of the above

11. Which of the following properties are obeyed by all three tree – traversals?

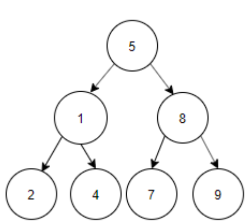
  a. Left subtrees are visited before right subtrees
  b. Right subtrees are visited before left subtrees
  c. Root node is visited before left subtree
  d. Root node is visited before right subtree

12. Suppose you have a binary tree with exactly 7 nodes. What is the maximum height that the tree could have?

  a. 2
  b. 3
  c. 6
  d. 7



13. Consider the following binary tree:



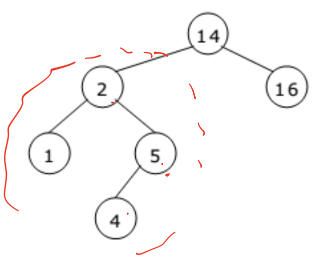
Which of the following statements is true?
I. It is a complete binary tree.
II. It is a strictly binary tree.
III. It is a binary search tree.

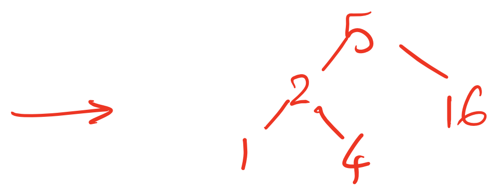
a. I only
b. I and II
c. I, II and III
d. I and III

14. Suppose node with key 7 is NOT THERE in the binary tree in the above question, then which of the following statements is true?
I. It is a complete binary tree.
II. It is a strictly binary tree.
III. It is a binary search tree.

a. I only
b. I only
c. III only
d. I and III
e. None of the statements is true.

15. For the binary search tree shown below, suppose the root is deleted and replaced with **a node in** the left subtree of the root, what will be the new root?

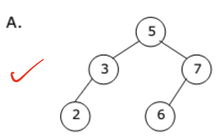




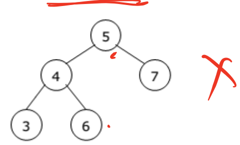
a. 1
b. 2
c. 4
d. 5

16. Which among the following is not a binary search tree?

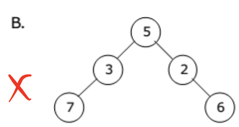
A.



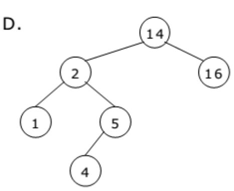
C.



B.



D.



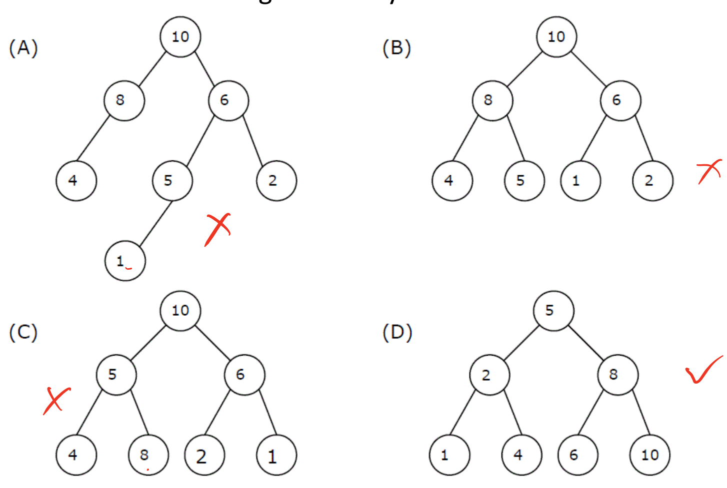
a. A and D
b. B only
c. C only
d. B and C

17. Which of the following statements is true for a binary search tree?

I. It must be a complete binary tree.
II. It must be a strictly binary tree.
III. The key x at a node must be larger than ~~or equal to~~ all the keys in the left subtree of the node.
IV. The key x at a node must be smaller than all the keys in the right subtree of the node.

a. I, II and III
b. I, II and IV
c. III only
d. III and IV

18. Which of the following is a binary search tree?



a. A
b. B
c. C
d. D
e. C and D

19. Given the following input (4322, 1334, 1471, 9679, 1989, 6171, 6173, 4199) and the hash function x mod 10, which of the following statements are true?

i. 9679, 1989, 4199 hash to the same value
ii. 1471, 6171 hash to the same value
iii. All elements hash to the same value
iv. Each element hashes to a different value

a. i only
b. ii only
c. iii only
d. iv only
e. i and ii

20. Given a hash table T with 25 slots that stores 2000 elements, the load factor α for T is

____

a. 80
b. 0.0125
c. 50,000
d. 1.25

$\text{Load factor} = \frac{\#\text{ keys}}{\text{table size}} = \frac{2000}{25} = 80$

21. A hash table of length 10 uses open addressing with hash function h(k)=k mod 10, and linear probing. After inserting 6 values into an empty hash table, the table is as shown below.

0	
1	
2	42
3	23
4	34
5	52
6	46
7	33
8	
9	

	a)		
0			
1			
2	42	42	42
3	52	23	23
4	34	34	34
5		52	52
6	46	33	46
7			33
8			
9			

Which one of the following choices gives a possible order in which the key values could have been inserted in the table?

a. 46, 42, 34, 52, 23, 33 ✗
b. 34, 42, 23, 52, 33, 46 ✗
c. 46, 34, 42, 23, 52, 33 ✓
d. 42, 46, 33, 23, 34, 52 ✗

22. A hash function h defined h(key)=key mod 10, with linear probing, is used to insert the keys 44, 45, 79, 55, 96, 15 into a table indexed from 0 to 9. What will be the location of key 15?

a. 5
b. 6
c. 7
d. 8

0	1	2	3	4	5	6	7	8	9
				44	45	55	96	15	79

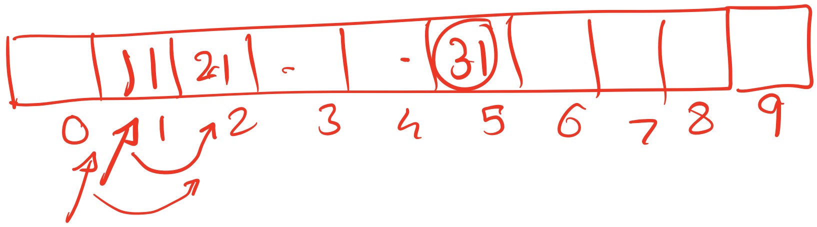
23. Suppose keys 19, 29, 39 and 49 are hashed into a hash table of size 10 in sequence with a hash function key % table_size, and closed hashing with linear probing is used, at what location will 49 be placed?

a. 0
b. 1
c. 9
d. 2

0	1	2	3	4	5	6	7	8	9
29	39	49							19

24. Suppose keys 11, 21, and 31 hashed into a hash table of size 10 in sequence with a hash function key % table_size, and closed hashing with quadratic probing is used, at what location will 31 be placed?

a. 2
b. 5
c. 1
d. 4



25. In cryptographic hashing, which of the following is correct?

a. The hash of the password is transmitted to the server and the server compares the hash with the stored value.

b. Hashing is not used for password checking.

c. The server sends the hash of the password to the user and the user compares the hash with the stored value.

d. The password is transmitted to the server and the server computes the hash of the password.

26. Consider an array with the following keys: [96, 51, 30, 27, 16, 19, 4]. Suppose it is being sorted in ascending order using Selection Sort. What will be the content of the array after **two iterations** of the algorithm?

a. [16, 4, 30, 27, 51, 19, 96]
b. [4, 16, 30, 27, 96, 19, 51]
c. [16, 4, 30, 27, 96, 19, 51]
d. [4, 16, 30, 27, 51, 19, 96]

96 51 30 27 16 19 4
1st: 4 51 30 27 16 19 96
2nd: 4 16 30 27 51 19 96

27. Consider an array with the following keys: [96, 51, 30, 27, 16, 19, 4]. Suppose it is being sorted in ascending order using Bubble Sort. What will be the content of the array after two iterations (two passes) of the algorithm?

a. [51, 30, 27, 16, 19, 4, 96]
b. [30, 27, 16, 19, 4, 51, 96]
c. [27, 30, 16, 19, 4, 96, 51]
d. [16, 27, 4, 51, 30, 19, 96]

96 51 30 27 16 19 4
1st Pass: 51 30 27 16 19 4 96
2nd Pass: 30 27 16 19 4 51 96

28. What is the worst-case complexity of sorting n keys with Merge Sort?

a. $O(n^2)$
b. $O(n \log_2 n)$
c. $O(n^2 \log_2 n)$
d. $O(n)$

29. Which of the following sorting algorithms have a worst-case complexity of $O(n^2)$, where n is the number of keys to be sorted?

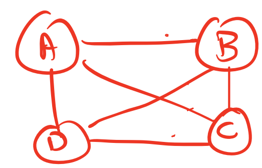
I. Bubble Sort
II. Selection Sort
III. Merge Sort
IV. Bucket Sort

a. I only
b. I and II
c. I, II and III
d. I, II, III and IV

30. An undirected graph G has four vertices. Its adjacency matrix representation has all 0's in its **MAIN** diagonal elements and all 1's in its non-diagonal elements. How many edges does the graph G have?

a. 3
b. 6
c. 8
d. 9

$$\begin{array}{c|cccc} & A & B & C & D \\ \hline A & 0 & 1 & 1 & 1 \\ B & 1 & 0 & 1 & 1 \\ C & 1 & 1 & 0 & 1 \\ D & 1 & 1 & 1 & 0 \end{array}$$



$$\frac{n(n-1)}{2} = \frac{4*3}{2} = 6$$

31. Which of the following is statements is correct with respect to graph representation?

I. Adjacency list representation is better than adjacency matrix for sparse graphs (graphs with large number of vertices and less number of edges)
II. Finding whether there is an edge between any two nodes in a graph is easier with Adjacency List than with Adjacency Matrix.
III. Finding all the edges of a given vertex is easier with Adjacency Matrix than with Adjacency List.

a. I only
b. I and II
c. I and III
d. II only
e. III only

32. Which of the following data structures is useful to traverse a graph using breadth first search?

a. Unordered List
b. Binary Tree
c. Stack
d. Queue

33. Which of the following would be the Depth First Traversal of the following graph?

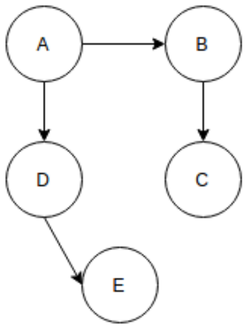


Figure 2

A B C D E
A D E B C

I. ABCDE
II. ADEBC
III. ABDCE
IV. ADBEC

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

a. I only
b. I and II
c. III only
d. III and IV

34. Consider the graph in Figure 2. How many zeros will be there in the adjacency matrix representation of the graph?

a. 4
b. 5
c. 21
d. 12

$25 - 4 = 21$

35. Consider the graph in Figure 2. The indegree and outdegree of vertex A are, respectively,

a. 1 and 1
b. 2 and 0
c. 0 and 2
d. 1 and 0

QUESTION NO. 2

Using the table, show the step by step process of how the binary search algorithm would work when you **search for the key 105** on the following array of integers. Note: not all the rows in the table may be filled.

	-5	-2	10	12	17	25	35	45	55	63	105
Index	0	1	2	3	4	5	6	7	8	9	10

lo	hi	mid	Key found?	Action
0	10	$\frac{0+10}{2} = 5$	105 > 25 NO	Go right
6	10	$\frac{6+10}{2} = 8$	NO 105 > 55	Go right
9	10	$\frac{9+10}{2} = 9$	NO 105 > 63	Go right
10	10	$\frac{10+10}{2} = 10$	yes!	

QUESTION NO. 3

For this question, assume that the Ordered List class has been developed and the methods in the class are as shown below:

```
Ordered List Methods
Constructor
OrderedList()                 Constructs an empty unordered list
Methods
int size()                    Returns size of the list
isEmpty()                     Returns true if list is empty
void clear()                  Clears the list
T get(int pos)                Gets the item at the specified index
T first ()                    gets the first entry
T next()                      gets the next entry
void enumerate()              scans the list and prints it
int binarySearch(T item)      searches for a given item.
                              returns position (index) if found
                              if not found returns a negative number
void insert(T item)           inserts the item at the correct position
void add(int pos, T item)     adds the given item at the specified index
void remove(T item)           remove the given item
void remove(int pos)          remove the item at the given index
```

Using the methods in this class, write a method that accepts an ordered list of Strings and a key and returns a new ordered list with all the elements (alphabetically) smaller than the key. For example, if the ordered list is: {A, C, F, G, Z} and the key is F, return the ordered list {A, C}

Note: The target key may or may not be in the ordered list.

Key is M

return {A, C, F, G}

Use the method header given below:

```
public static OrderedList<String> subList(OrderedList<String> list1, String k1)
{
```

```
OrderedList<String> result= new OrderedList<String>();
if (list1.isEmpty()) return result;
int i= 0;
String check = list1.first();
while ((i < list1.size()) && (check.compareTo(k1)<0))
{
      result.add(check);
      check = list1.next();
}     i++;
   return result;
}
```

QUESTION NO. 4 (RECURSION)

4.1 Consider the following recursive method:

```
public static void showMe(int n){
      if (n>10)
            System.out.print(n);
      else
            showMe(n+3);
      }
}
```

showMe(5)
↓
showMe(8)
↓
showMe(11)

Answer: 11

Assuming that the main method has been implemented correctly, what will be displayed if the main method calls showMe(5)?

4.2

```
public static void showMe(int n){
      if (n<10){
(1)         System.out.print(n + " ");
(2)         showMe(n+3);
      }
      else
            return;
}
```

showMe(5)
↓
showMe(8)
↓
showMe(11)

Answer: 5 8

Assuming that the main method has been implemented correctly, what will be displayed if the main method calls showMe(5)?

What is the output if you switch lines (1) & (2)

Answer: 8 5

4.3 Suppose that a recursive method implements the recursive definition given below:
func (0) = 0
func(N) = func(N-2) + N
Assuming that the definition has been implemented correctly, what will be displayed if the main method has a statement System.out.println(func(8))?

func(8) 20
func(6) + 8 12
func(4) + 6 6
func(2) + 4 2
func(0) + 2 0
0

Answer: 20

4.4 Consider the following recursive method:

```
public static int foo(int a, int b){

                if (b<=1 || b<=a)
                        return 1;
                else
                        return (b-a) * foo(a, b-1);
}
```

Assuming that the main method has been implemented correctly, what will be displayed if the main method has a statement System.out.println(foo(5,9))?

a b
foo (5, 9)
24 4 * foo(5, 8)
6 3 * foo(5, 7)
2 2 * foo(5, 6)
1 1 * foo(5, 5)
1 1

Answer: 24

4.5 Consider the following recursive method:

```
public static int mystery(int x, int n){
          if (n>0)
               return (mystery(x, n-1)*x);
          else
               return 1;
}
```

Assuming that the main method has been implemented correctly, what will be displayed if the main method has a statement System.out.print(mystery(2,3))?

Answer: 8

2^3 or x^n

mystery(2, 3) → 8

mystery(2, 2) * 2 → 4

mystery(2, 1) * 2 → 2

mystery(2, 0) * 2 → 1

1

4.6 Write a recursive method public static int count(int n) { ....} to count the number of digits in a given positive integer. For example, count(3456) should return 4, count(901101) should return 6, and count(4) should return 1.
public static int count(int n){
// continue

```
if (n/10 == 0)
        return 1;

else
    return (1 + count(n/10));

}
```

$\frac{3456}{10} \rightarrow 345$

$\frac{345}{10} \rightarrow 34$

$\frac{34}{10} \rightarrow 3$

$\frac{3}{10} \rightarrow 0 \rightarrow$ exit

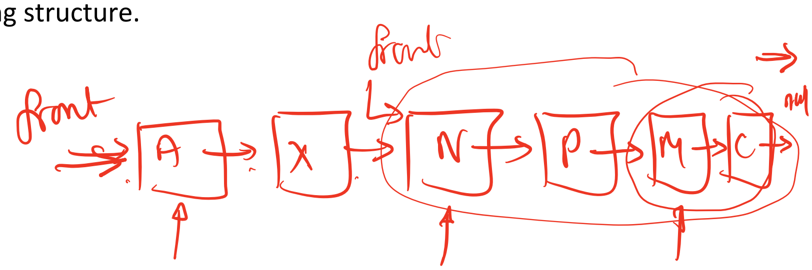
4.7: Write a recursive method for displaying the data in the even-numbered nodes (0, 2, 4, etc.) in a linked list. Assume that the nodes are numbered from 0. For example, if the linked list is → A→X→N→P→M→C, it should display

A   N   M

Assume that the LinkedList<T> class has the following structure.

```
public class LinkedList<T>
{
        private Node<T> front;
        private int count;

        public LinkedList()
        {
                front = null;
                count=0;
        }
```



You may also use the methods in the Node class.

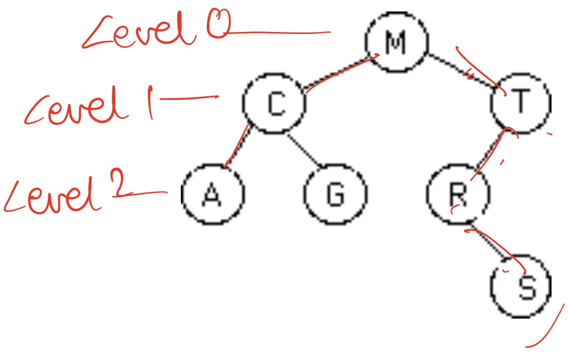
```
public void displayEven (Node<T> front)
{
    if (front == null)
            return;
    else if (front.getNext() == null)
            System.out.print(front.getData() + " ");
    else {
            System.out.print(front.getData() + " ");
            displayEven(front.getNext().getNext());
    }
}
```

A   N   M

A   N   M

QUESTION NO. 5

You are given the following binary tree:



i) Which are the leaf nodes? A, G, S

ii) Which are the internal nodes? M, C, T, R

iii) What is the longest distance (between any two nodes) in the tree? 5 (A → S)

iv) Which are the nodes at Level 2? A, G, R

v) What is the height of the tree? 3

vi) What are the leaf nodes in the right subtree of the root node? S

vii) What is the left subtree of the left child of the right child of the root node? Null

viii) Is it a strictly binary tree? Why or why not? No. T & R have only one child

ix) Is it a complete binary tree? Why or why not? No. Level 2 & 3 are not full & not ordered.

QUESTION NO. 6 (BINARY TREE SHORT SNAPPERS)

6.1 What is the maximum number of comparisons required by a binary search algorithm on an array of size 16? In general, if the number of items n in the array is a power of 2, what is the maximum number of comparisons?

$16 \rightarrow 4 + 1$ searches

2^4 $\quad \log_2 16 + 1$ searches

2^n $\quad \log_2 n + 1$ searches

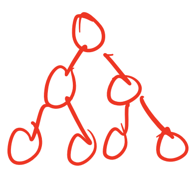
6.2 What is the maximum number of comparisons required by a binary search algorithm on an array of size 20? In general, if the number of items n in the array is not a power of 2, what is the maximum number of comparisons?

$20 \rightarrow 4 + 1$ searches $\quad 20, 10, 5, 2, 1$

$\lfloor \log_2 20 \rfloor + 1$ searches $\quad \lfloor \log_2 n \rfloor + 1$ searches

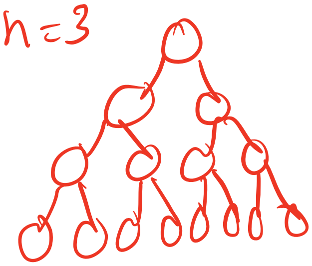
6.3 What is the maximum number of nodes in a complete binary tree of height = 3? In general, what is the maximum number of nodes in a complete binary tree of height h?

$h = 2$



7

$h = 3$

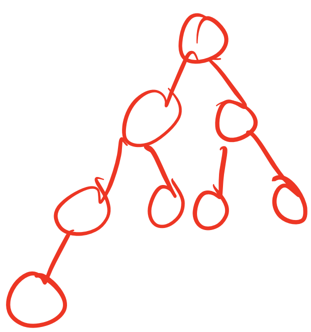


15

$2^{h+1} - 1$

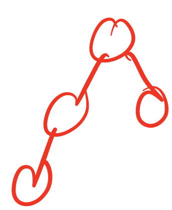
6.4 What is the minimum number of leaves in a complete binary tree of height = 3? In general, what is the minimum number of nodes in a complete binary tree of height h?

$h = 3$



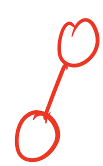
leaves = 4

$h = 2$



2

$h = 1$

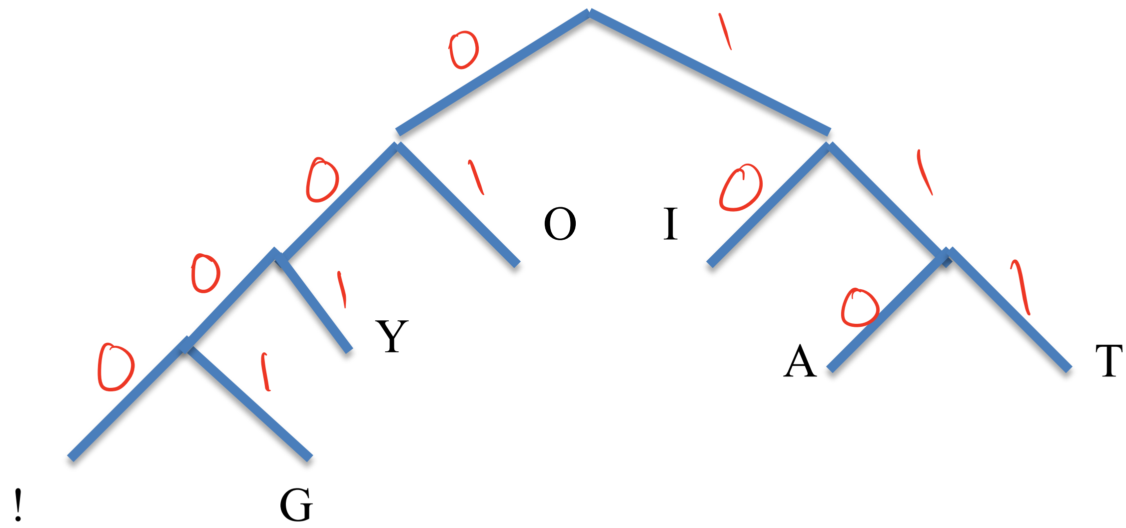


1

2^{h-1} leaves

QUESTION NO. 7

The following tree was generated using the Huffman technique.



The following message was generated using the codes from the above tree.

0011100010000 **10** **000101111** **1011100000000**

Decode the message. Leave the spaces as they are.

YAY! I GOT IT!!

QUESTION NO. 8

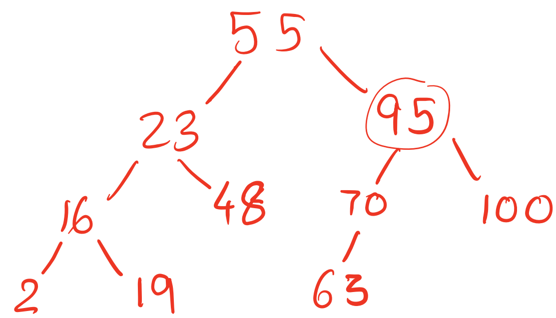
Write a recursive method to count the number of nodes in a binary tree. Assume that the method is implemented within the Binary Tree class.

```
public int countNodes(BinaryTree<T> tree)
{
    if (tree == null)
        return 0;
    else if (tree.getLeft() == null &&
             tree.getRight() == null)
        return 1;
    else
        return (1 + countNodes(tree.getLeft())
                  + countNodes(tree.getRight()));
}
```

QUESTION NO. 9

Starting from an empty **binary search tree,** suppose that the following operations are performed ***in sequence***.

- Insert 55, 23, 48, 95, 16, 2, 70, 63, 100, 19



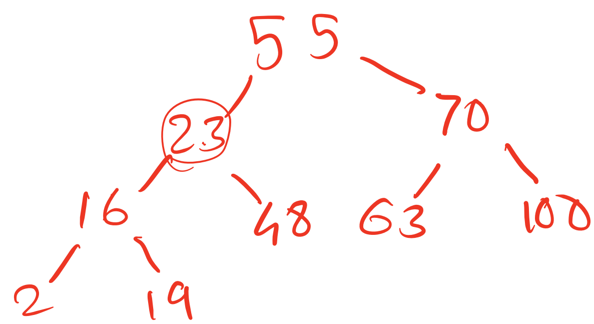
a) What is the root node?

b) What are the leaf nodes?

c) What are the left and right child nodes of the root?

a) 55
b) 2, 19, 48, 63, 100
c) 23, 95

- Delete the right child of the root node

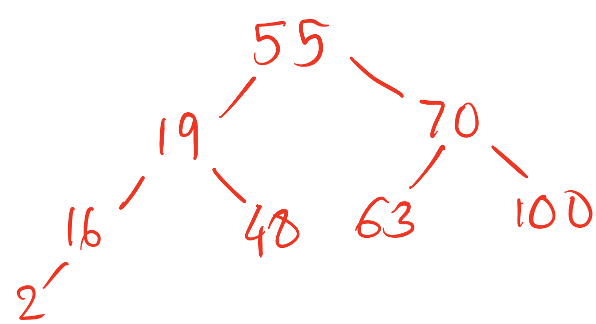


a) What are the leaf nodes?

b) What are the left and right child nodes of the root?

a) 2, 19, 48, 63, 100
b) 23, 70

- Delete the left child of the root node



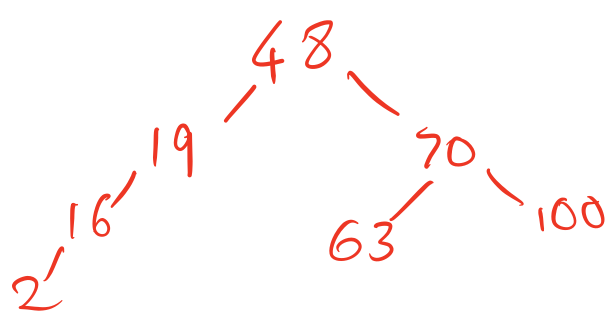
a) What are the leaf nodes?

b) What are the left and right child nodes of the root?

a) 2, 48, 63, 100
b) 19, 70

- Delete the root node

a) What is the new root node?



b) What are the leaf nodes?

c) What are the left and right child nodes of the root?

a) 48
b) 2, 63, 100
c) 19, 70

QUESTION NO. 10 (HASHING)

10.1 The keys 12, 18, 13, 2, 3, 23, 5 and 15 are inserted into an initially empty hash table of length 10 using closed hashing and linear probing. The hash function is h(k) = k mod 10.

What is the resultant hash table? Write your answer as a sequence of numbers separated by commas and a '-' for an empty array slot.

For example, if your resulting hash table is the array (Note: This is not the correct answer!)

12		13		18	2	3	23	5	15

You would write it your answer as follows:
12, - , 13, - , 18, 2, 3, 23, 5, 15

0	1	2	3	4	5	6	7	8	9
		12	13	2	3	23	5	18	15

Answer: –, –, 12, 13, 2, 3, 23, 5, 18, 15

10.2 Repeat the above question but this time use quadratic probing.

0	1	2	3	4	5	6	7	8	9
		12	13	3	5	2	23	18	15

Answer: –, –, 12, 13, 3, 5, 2, 23, 18, 15

10.3 Consider a hash table of size seven, with starting index zero, and a hash function (3x + 4) mod 7, where x is the key. Assuming the hash table is initially empty, the sequence of keys 1, 3, 8, 10 is inserted into the table using closed hashing and linear probing?

What is the resultant hash table? Write your answer as a sequence of numbers separated by commas and a '-' for an empty array slot.

0	1	2	3	4	5	6
1	8	10				3

$$1 \quad (3*1+4) \bmod 7 = 0$$
$$3 \quad (3*3+4) \bmod 7 = 6$$
$$8 \quad (3*8+4) \bmod 7 = 0$$
$$10 \quad (3*10+4) \bmod 7 = 6$$

Answer:
1, 8, 10, –, –, –, 3

10.4 The figure below shows the current state of a 10-element hash table, with the occupied cells shown with an XX. Assuming that the hash function is key mod 10. The following keys are inserted in sequence:

44, 55, 66, 77

0	1	2	3	4	5	6	7	8	9
	XX		XX	XX	XX	XX	XX		

a) What is the resulting table if Linear probing is used? What is the total number of collisions?

b) What is the resulting table if Quadratic probing is used? What is the total number of collisions?

0	1	2	3	4	5	6	7	8	9
66	X	77	X	X	X	X	X	44	55

collisions = 17

44 → 4
55 → 4
66 → 4
77 → 5

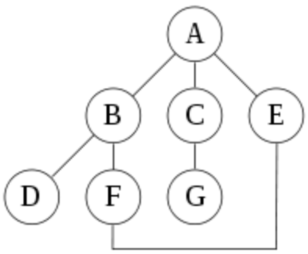
0	1	2	3	4	5	6	7	8	9
66	X	77	X	X	X	X	X	44	55

collisions = 11

44 : 1+1
55 : 1+1
66 : 1+1
77 : 5

QUESTION NO. 11:

For the graph given below
a) State whether the graph is weighted or unweighted, directed or undirected, connected or unconnected, cyclic or non-cyclic.
b) List all vertices with degree = 3.
c) List a cycle in the graph, if it exists.
d) List all possible simple paths from A to D.
e) Determine the DFS and BFS traversals from node A.



a) Unweighted, Undirected, Connected, Cyclic

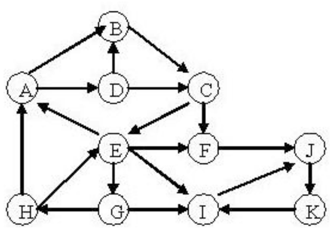
b) A, B

c) A E F B A

d) ABD, AEFBD

e) ABDFECG → DFS ; - ABCEDFG → BFS

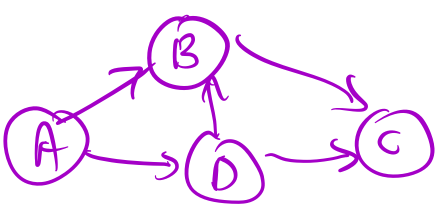
QUESTION NO. 12:
For the following graph, which node(s) has (have) the highest indegree? What is the highest indegree? Which node(s) has (have) the highest outdegree? What is the outdegree? Is the subgraph connecting vertices A, B, C, D strongly connected or weakly connected?



Answers:

Highest indegree: I 3

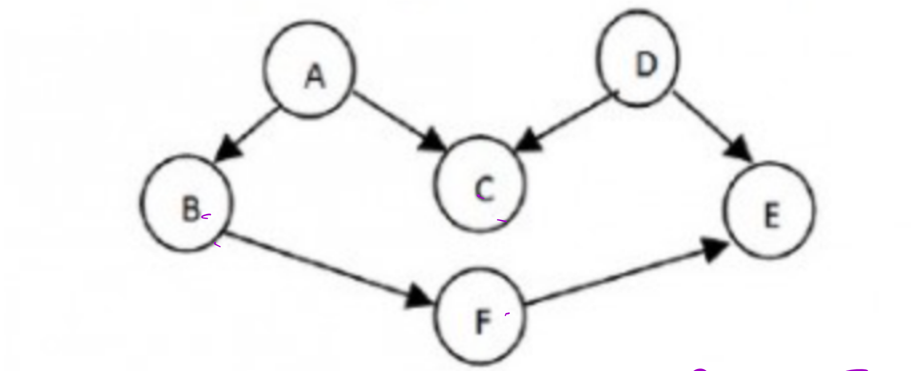
Highest outdegree: E 4



Weakly connected.

QUESTION NO. 13

For the graph shown below, find one solution that lists the vertices so that the dependency is not violated. (Run the topological sorting algorithm and assign topological numbers to the vertices. Show steps).



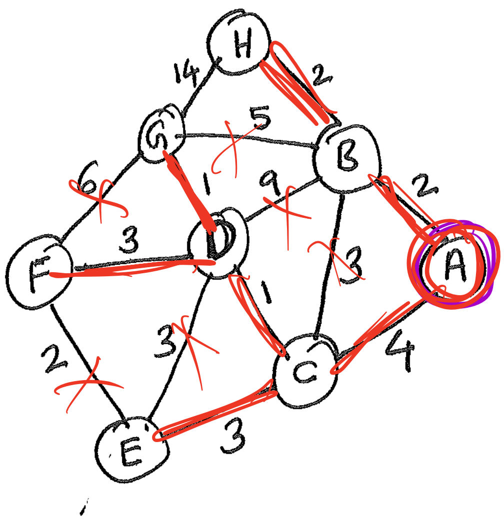
	A	B	C	D	E	F
PRED	0	~~1~~ 0	~~2~~ ~~1~~ 0	0	~~2~~ ~~1~~ 0	~~1~~ 0

QUEUE	~~A~~	~~D~~	~~B~~	~~C~~	~~F~~	~~E~~

TOPNUM	1	2	3	4	5	6
	A	D	B	C	F	E

QUESTION NO. 14

Using Dijkstra's algorithm, find the shortest paths from node A to all other nodes in the following graph, Show steps.



Adjacency List

A	B	C	D	E	F	G	H
B,2	A,2	A,4	B,9	C,3	D,3	B,5	B,2
C,4	C,3	B,3	C,1	D,3	G,6	D,1	G,14
	D,9	D,1	E,3	F,2	E,2	F,6	
	G,5	E,3	F,3			H,14	
	H,2		G,1				

Sl.No.	C list	T list	Remarks
1	A, 0, -	B, 2, B C, 4, C	Moved A to C list. Its neighbours to T list
2	B, 2, B	C, 4; C D, 11, B G, 7, B H, 4, B	Move B to C list Retain old entry for C Add entries for D, G & H
3	C, 4, C	D, 5, C E, 7, C G, 7, B H, 4, B	Move C to C list. New entry for D Add entry for E
4	H, 4, B	G, 7, B D, 5, C E, 7, C	Move H to C list Retain old entry for G
			Move D to C list

5	D, 5, C	E, 7, C F, 8, C G, 6, C	Retain old entry for E Add entry for F New entry for G
6	G, 6, C	F, 8, C E, 7, C	Move G to C list Retain old entry for F
7	E, 7, C	F, 8, C	Move E to C list Retain old entry for F
8	F, 8, C	—	