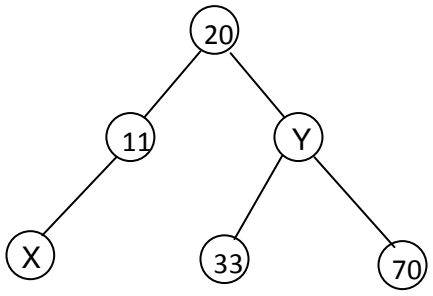
1) Given the following binary search tree:



Which of the following best represents the possible values for X and Y if duplicates are not allowed?

A. $X < 11$ and $20 < Y < 70$

B. $11 < X < 20$ and $33 < Y < 70$

C. $X < 11$ and $33 < Y < 70$

D. $11 < X < 20$ and $Y < 20$

E. $X < 11$ and $Y > 33$

2) You are given an array of size 100 where the value at each index k is k, e.g., A[0] = 0, A[1] = 1, etc. You perform a search for every value in the array, e.g., first you find 0, then 1, etc. The algorithms used to search for a value use the general techniques discussed in lecture and the on-line readings. Which of the statements below is *true*?

A. Binary search is not applicable to this problem.

B. Sequential search is not applicable to this problem.

C. Sequential search is *always faster* than binary search for *every* one of the 100 values searched.

D. Sequential search is *always slower* than binary search for *every* one of the 100 values searched.

E. Sequential search is *sometimes faster, sometimes slower* than binary search *depending* on which of the 100 values is being searched.

The next two questions refer to the following method:

```
public static int compute(int x, int y) {
    if (x == y) return x;
    else        return compute(x + 1, y - 1);
}
```

3) What is returned by the call `compute(1, 5)`?

A. 1
B. 2
C. 3
D. 4
E. No value is returned because an infinite recursion occurs.

4) Which of the following calls leads to an infinite recursion?

i. `compute (2, 8)`

ii. `compute (8, 2)`

iii. `compute (2, 5)`

A. i only B. ii only C. iii only D. i and ii E. ii and iii

5) Assume that general trees are implemented using a `Treenode` class that includes the following fields and methods:

```
private List<Treenode<T>> children;

private List<Treenode<T>> getChildren() { return children; }
```

and that the following class is used to represent general trees:

```
public class Tree<T> {
    Treenode<T> root;
    // other methods will come here
}
```

Consider the following methods (assume they are in the `Tree` class):

```
public int count() { return count(root); }

private int count(Treenode<T> n) {
    if ((n == null) || n.getChildren().isEmpty())
        return 0;
    int total = 1;
    Iterator<Treenode<T>> iter = n.getChildren().iterator();
    while (iter.hasNext())
        total += count(iter.next());
    return total;
}
```

Which of the following best describes what the method `count` does?

A. Always returns 0.

B. Returns the number of nodes in the tree.

C. Returns the number of leaves in the tree.

D. Returns the number of non-leaves in the tree.

E. Returns the height of the tree.

1) Assume that binary trees are implemented using a `BinaryTreenode` class that includes the following fields and methods:

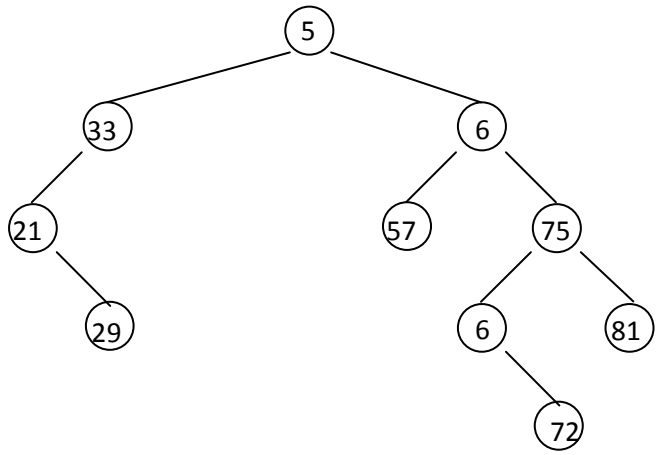
```
// fields
private T data;
private BinaryTreenode<T> left, right;

// methods
public T getData()     { return data; }
public BinaryTreenode<T> getLeft()  { return left; }
public BinaryTreenode<T> getRight() { return right; }
public void setLeft(BinaryTreenode<T> newL)  { left = newL; }
public void setRight(BinaryTreenode<T> newR) { right= newR; }
```

Write the `countTwoChildren` method whose header is given below. The method should return the number of nodes in the tree that have exactly two children. To receive full credit you must use recursion in your solution. You may not add any additional data members or methods to the `BinaryTreenode` class.

```
public int countTwoChildren(BinaryTreenode<T> N) {
```

2) Given the following binary search tree:



Part a) Give the order in which the nodes would be printed using the given traversal. Note: children should be visited from left to right.

Pre-order traversal:

Post-order traversal:

In-order traversal:

Level-order traversal:

Part b) Show the tree that results from inserting the value 27, deleting the value 33, and deleting the value 60 (in that order). Deletion should use the inorder successor (where appropriate).

3) Assume that `s` is a non-empty string containing only digits. Consider the following recursive implementation to convert the string to the corresponding integer value (e.g., `convertToInt("1234")` returns the integer 1234):

```
public static int convertToInt(String s) {

   // determine the integer value of the last character in s
   int value = s.charAt(s.length() - 1) - '0';

   // if s only has one character, return its integer value
   if (s.length() == 1) return value;

   // otherwise the integer value of s is
   //   (integer value of s without its last character)*10
   //    + integer value of the last character in s
   return convertToInt(s.substring(0, s.length() - 1))*10 + value;
}
```

Analyze the time complexity for the recursive `convertToInt` method above:

Part a) Identify the problem size (what affects the method's runtime),

N =

Part b) Write the recurrence equations for base and recursive cases,

T(1) =

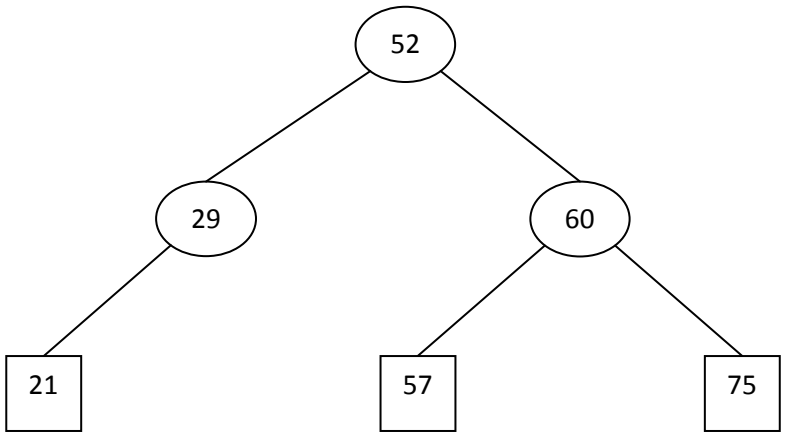
T(N) =

Part c) State your solution that you've guessed by looking for a pattern, and

Part d) Verify your solution.

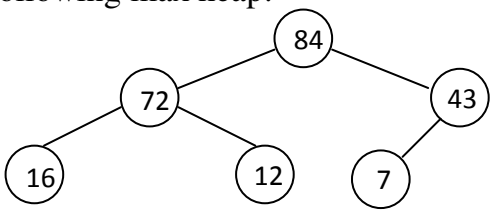
Part e) What is the complexity (in big-O notation) of the `convertToInt` method?

4) Given the following red-black tree (where black nodes are depicted using ovals and red nodes are depicted using squares):



Show the tree that results from inserting the value 25, 54, and 80 (in that order).

5) Given the following max heap:



Show the heap (in tree form, as above) that results from performing one `removeMax` operation followed by inserting the value 56.

Part I True or False *[12 questions, 2 points each, 24 total points]*

For questions 1 through 12, is the statement true or false? Mark A for true or B for false.

1.) A complete, undirected graph implemented by storing edges in adjacency lists will be a significantly more efficient use of space than if the edges are stored in an adjacency matrix.

2.) A min-heap is a good way to implement a priority queue if the highest priority has the smallest integer value.

3.) The efficiency of quick sort is $O(n^2)$ in the worst case, but its expected efficiency is $O(n \log_2 n)$ because the worst case can almost always be avoided.

4.) A tree can be used to store the vertices and edges of any directed acyclic graph.

5.) AVL trees keep their balance by splitting the root node when an insert results in one subtree being more than one level greater than another subtree of the same node.

6.) A 2-3 tree that contains many values is likely to have fewer levels than the red-black tree with the same values, and is likely to have more levels than the 2-3-4 tree with the same values.

7.) A full binary search tree has $\log_2 (h - 1)$ nodes where h is the height of the tree. Assume that a one node tree has a height of one.

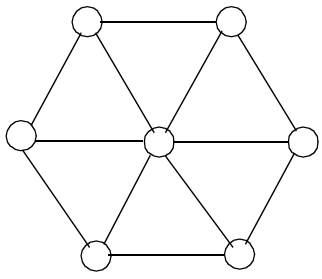
8.) An array can be used to store the data of 5-ary trees.

9.) Siblings in a Tree ADT have connections between them.

10.) Assume a program needs to store information about the cost of gas at several gas stations. A good ADT choice would be a weighted graph, where the vertices represent the gas stations and the edges represent the cost of gas.

11.) A complete binary tree has at least $2^{(h-1)} - 1$ nodes where h is the height of the tree. Assume that a one node tree has a height of one.

12.) The diagram shown below is a complete graph.



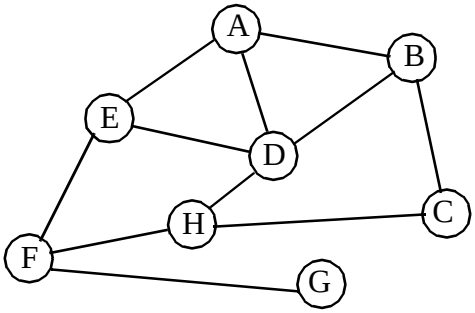
Part II Multiple Choice *[14 questions, 4 points each, 56 total points]*

For questions 13 through 26, choose the best answer of the given possibilities.
Mark the corresponding letter on your answer sheet.

13.) Which one of the following statements is **False**?

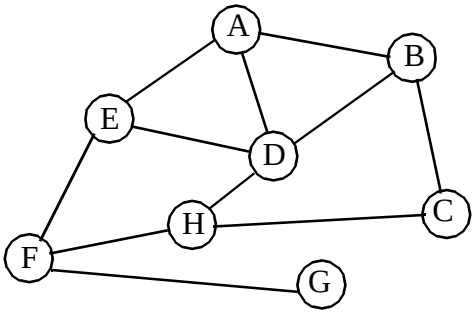
A. Trees are special types of graphs.
B. Trees are non-linear data structures.
C. Stacks are special types of lists.
D. Stacks are linear data structures.
E. Deques are non-linear data structures.

14.) Which list of vertices represents a possible depth-first traversal of this graph?



A. starting at A: A D E F G H C B
B. starting at B: B D E F G H A C
C. starting at C: C H B G E F D A
D. starting at D: D A E F G H B C
E. starting at E: E D A F G H C B

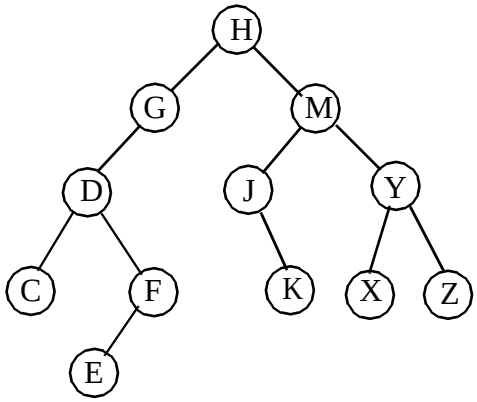
15.) Which list of vertices represents a possible breadth-first traversal of this graph?



A. starting at A: A B E D G F C H
B. starting at B: B D A C H F G E
C. starting at C: C H B G E F D A
D. starting at D: D A E H B F C G
E. starting at E: E D F G H A B C

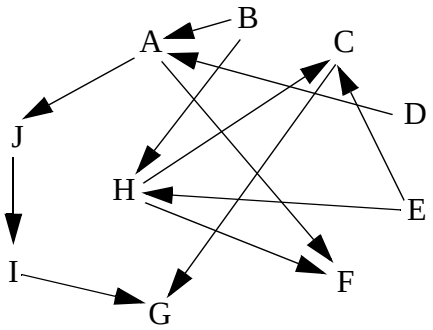
19.) Which choice correctly shows the contents of the array that represents this binary search tree after removing "H"? Assume the same array representation and remove algorithm as required in A4. Use recursion to replace any non-leaf node being removed with greatest of its left subtree if possible, else replace the node with the least of its right subtree. In the array, the first element of the array is left unused and is shown as the underscore character. Trailing, unused elements of the array are not shown.

The array contents before removing H are: _, H, G, M, D, _, J, Y, C, F, _, _,_, K, X, Z, _, _, E



A. _, E, G, M, D, _, J, Y, C, F, _, _, _, K, X, Z
B. _, F, G, M, D, _, J, Y, C, E, _, _, _, K, X, Z
C. _, G, F, M, D, _, J, Y, C, E, _, _, _, K, X, Z
D. _, G, D, M, E, _, J, Y, C, F, _, _, _, K, X, Z
E. _, G, D, M, C, F, J, Y, _, _, E, _, _, K, X, Z

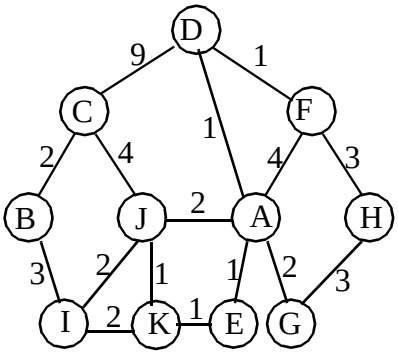
20.) What is a valid topological order of this graph?



A. No topological order is possible or is not one of the following choices.
B. D B A E H C J I G F
C. G F H I J C E A B D
D. B A H C G J I D E H
E. D B A H J I E C F G

25.) Which choice lists all vertices that have incorrect rows in the adjacency matrix for this graph? Note: rows are horizontal in this matrix.

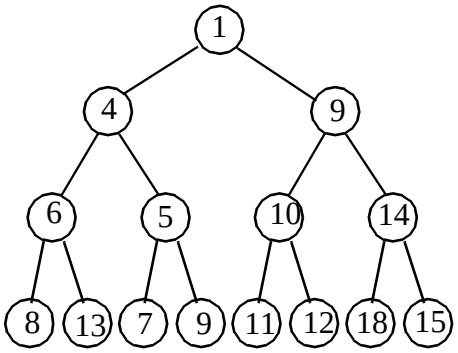
	A	B	C	D	E	F	G	H	I	J	K
A				1	1	4	2			2	
B			2						4		
C		2		9						4	
D	1		9			1					
E	1								1		1
F	4			1		1		3			
G	2							3			
H						3	3				
I		3			1					2	2
J	2		4						2		1
K					1					1	



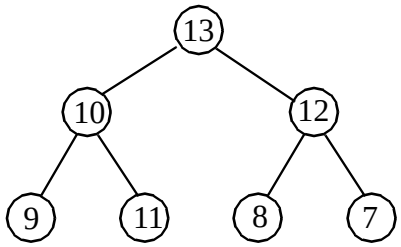
A. I
B. I K
C. B
D. B K
E. B E F I K

26.) Which of the following is a maxheap of integer values?

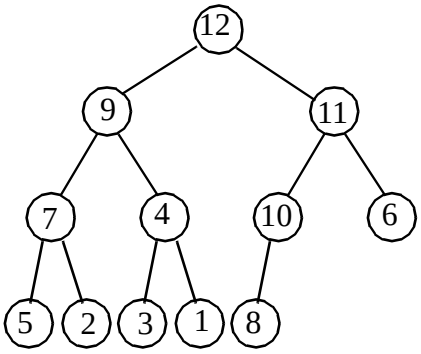
A



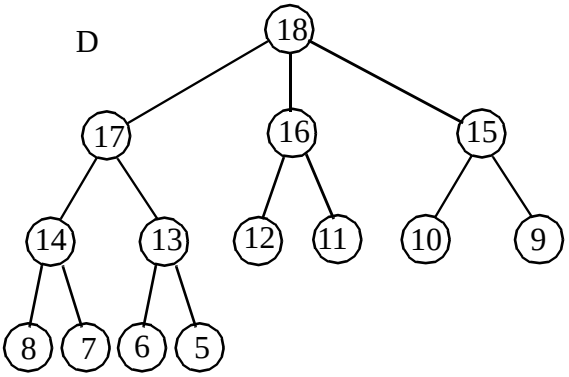
B



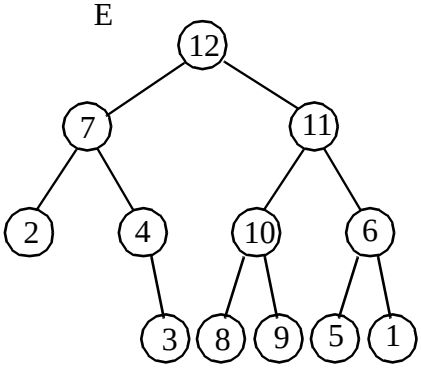
C



D



E



Part III Written Answers *[3 questions, 19 total points]*

Write your answers to the remaining questions in this examination booklet. If you need more room indicate where your work continues.

27.)[6 points] Fill in the following table with tightest O( ) for the method's execution time for the specified ADT. Give your answer in terms of n, where n is the number of items.

Complete for ADT: **Binary Search Tree**		
Operation	**array**	**linked node**
`look up`		
`add`		
`remove`		

28.)[6 points] The Wing and a Prayer Airline Company cannot keep up with quote requests for ticket prices, because the algorithm their computer program uses to determine fares is $O(n^3)$. They fly to thousands of cities, and a weighted graph of their routes gives fare prices for each leg of a flight. They have offered free airline tickets to anyone who can help them. How do you get those free tickets? Justify it in terms of O( ) efficiency.