UNIVERSITY OF COLOMBO, SRI LANKA

UNIVERSITY OF COLOMBO SCHOOL OF COMPUTING

DEGREE OF BACHELOR OF INFORMATION TECHNOLOGY (*EXTERNAL*)
Academic Year 2003/2004 - 1st Year Examination – Semester 2

IT2202 – Data Structures & Algorithms

14th August, 2004
(TWO HOURS)

Important Instructions :

The duration of the paper is 2 (two) hours.

The medium of instruction and questions is English.

The paper has **45** questions and **16** pages.

All questions are of the MCQ (Multiple Choice Questions) type.

All questions should be answered.

Each question will have 5 (five) choices with **one or more** correct answers.

All questions will carry equal marks.

There will be a penalty for incorrect responses to discourage guessing.

The mark given for a question will vary from –1 (*All the incorrect choices are marked & no correct choices are marked*) to +1 (*All the correct choices are marked & no incorrect choices are marked*).

Answers should be marked on the special answer sheet provided.

Note that questions appear on both sides of the paper.
If a page is not printed, please inform the supervisor immediately.

Mark the correct choices on the question paper first and then transfer them to the given answer sheet which will be machine marked. **Please completely read and follow the instructions given on the other side of the answer sheet before you shade your correct choices.**

1) Consider the following statements.

(i) A stack is a list with the restriction that items are inserted or removed/deleted only at one position, namely the end of the list.
(ii) The general model is that one where there is some element that is at the top of the stack, and it is the only element that is visible.
(iii) A pop or top on an empty stack is generally considered an error in the stack ADT.
(iv) The fundamental operations on stacks are push and pop, where push is relevant to the removal of the most recently inserted element and pop is equivalent to an insertion.
(v) A stack is a list; insertion and deletion can be performed from both ends.

Which of the above statements is/are valid for stacks?

(a) (i), & (ii) only	(b) (i), (ii) & (iv) only	(c) (i), (ii), (iii) & (iv) only
(d) (i), (ii) & (iii) only	(e) (i), (ii) & (v) only	

2) Consider the following algorithm.

(i) Make an empty stack.
(ii) Read characters until end of the file.
(iii) If the character is an opening symbol, push it into the stack.
(iv) If it is a closing symbol, then if the stack is empty, report an error.
(v) Otherwise, pop the stack. If the symbol popped is not the corresponding opening symbol, then, report an error.
(vi) At end of the file, if the stack is not empty, report an error.

Identify, what the above algorithm intends to do.

(a) Evaluating a mathematical expression
(b) Check for balancing of parentheses, brackets and braces and ignore any other character that appears
(c) Eliminating un-matched parentheses, brackets and braces and ignoring any other character that appears
(d) Checking compiler errors
(e) Determination of syntax errors

3) Consider the Java program segment given below.

```
public object abc( ) throws undereflow
{
if (isempty( ))
throw new underflow (" stack underflow");
object topitem=topofstack.element;
topofstack=topofstack.next;
return topitem;
}
```

Which of the following describes the activities of the above Java program segment?

(a) Exception underflow if stack is empty
(b) Exception overflow is stack is full
(c) Remove the most recently inserted item from the stack
(d) Remove the highest priority item from the stack
(e) Return the most recently inserted item from the stack

4) Consider the following infix expression.

((A+B)*C-(D-E))^(F+G)

Note: ^ denotes the power.
Equivalent postfix and prefix expressions are respectively

(a) AB+C*DE--FG+^ , ^-*ABC-DE+FG	(b) AB+C*DE-FG+^ , ^-*+ABC-DE+FG
(c) AB+C*DE--FG+^ , ^-*+ABC-DE+FG	(d) AB+C*DE-FG+^ , ^-*+ACB-DE+FG
(e) AB+C*DE--FG+^ , ^-*+CBA-DE+FG	

5) Consider the following Java program segment.

```
BinaryNode  duplicate( )
{
BinaryNode root = new BinaryNode (Element);
if (left !=null)
root.left=left.duplicate( );
if (right !=null)
root.right=right.duplicate( );
return root;
}
```

What is the above Java program segment intended to do?

(a) Merge routine for the binary search class
(b) Routine to create a duplicate Node
(c) Finding duplicate nodes from the binary search tree
(d) Routine to return a copy of the tree rooted at the current node
(e) Removing duplicates from the binary search tree

6) Consider the following statements.

(i) A linked list consists of a series of structures, which are necessarily adjacent in memory.
(ii) In a singly linked list, each structure contains an element and a reference to a record containing its successor.
(iii) In an array-based list, even if the array is dynamically allocated, an estimate of the maximum size of the list is required.
(iv) In an array based list, inserting at position 0 requires first pushing the entire array down one spot to make room.
(v) In an array-based list, deleting elements from the middle can be performed without shifting the remaining elements.

Which of the above statements is/are valid for a list?

(a) (ii) & (iv) only	(b) (ii), (iii) & (iv) only	(c) (iii), (iv) & (v) only
(d) (ii) & (iv) only	(e) (ii), (iii), (iv) & (v) only	

7) Which of the following is/are **not** (a) valid queue application(s)?

(a) When printing jobs are submitted to printer
(b) Lines at tickets counters
(c) Evaluating a mathematical expression
(d) Calls to large companies, when all lines are busy
(e) When there are many network set-ups of personal computers in which the disk is allocated to one machine, known as the file server and users on the other machine are given access to the file

8) Consider the following activities.

(i) Remove and return the least recently inserted item from the queue, or return null if empty.
(ii) Set the next item as the least recently inserted item.

Which of the following Java code perform(s) the above activities?

(a)
```
public Object aaa( )
{
if (isempty ( ) )
return null;
currentSize--;
Object frontItem = theArray[ front];
theArray[front] =null;
front++;
return frontItem;
}
```

(b)
```
public Object aaa( )
{
if (isempty ( ) )
currentSize--;
Object frontItem = theArray[ front];
theArray[front] =null;
front++;
return frontItem;
   }
```

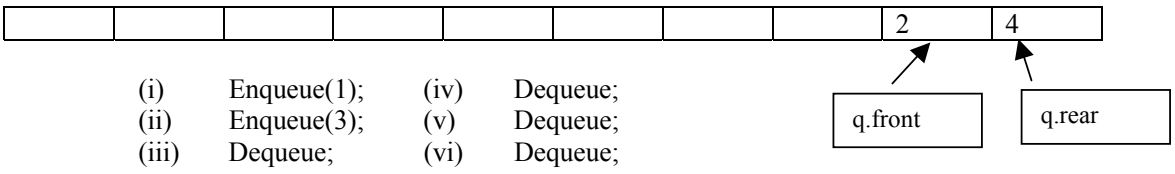
(c)
```
public Object aaa( )
{
if (isempty ( ) )
return null;
currentSize--;
Object frontItem = theArray[ front];
theArray[front] =null;
front++;
}
```

(d)
```
public Object aaa( )
{

Object frontItem = theArray[ front];
theArray[front] =null;
front++;
return frontItem;
}
```

(e)
```
public Object aaa( )
 {
if not  (isempty ( )
Object frontItem = theArray[ front];
theArray[front] =null;
front++;
return frontItem;
}
```

9) Consider the following queue with the indicated initial states and series of queue operations.



(i) Enqueue(1);
(ii) Enqueue(3);
(iii) Dequeue;
(iv) Dequeue;
(v) Dequeue;
(vi) Dequeue;

If the above series of operations is performed, what is the final status of the queue?

(a)

	q.front q.rear								

(b)

q.front	q.rear								

(c)

q.rear	q.front								

(d)

	q.rear	q.front							

(e)

	q.front	q.rear							

10) Consider the following java program segment.

```
   package datastructures;                          (i)
    class ListNode
{
//constructors
ListNode ( object theElement )                      (ii)
{
this (theElement,null );
}
ListNode ( Object theElement, ListNode n)           (iii)
{
element = theElement;
next  = n;
}
 Object element;                                    (iv)
  ListNode next;
}
```

Which of the following is/are the complete Java type declaration(s) for the linked list node?

(a) (i) & (iv) only	(b) (i), (ii), (iii) & (iv) only	(c) (i), (ii) & (iv) only
(d) (i), (iii) & (iv) only	(e) (ii), (iii) & (iv) only	

11) Consider the following operations.

(i) Append an element to the end of a list.
(ii) Concatenate two lists.
(iii) Free all the nodes in a list.
(iv) Reverse a list, so that the last element becomes the first and so-on.
(v) Delete the last element from a list.
(vi) Delete the n^{th} element from a list with at least n elements.
(vii) Combine two ordered lists into a single ordered list.

Which of the above are valid operations in singly linked lists?

(a) (i), (ii), (iii), (v), (vi) & (vii)	(b) (iii), (iv), (v), (vi) & (vii)
(c) (i), (ii), (iii), (iv), (vi) & (vii)	(d) (i), (ii), (iii), (iv) & (vii)
(e) (i), (ii), (iii), (iv), (v), (vi) & (vii)	

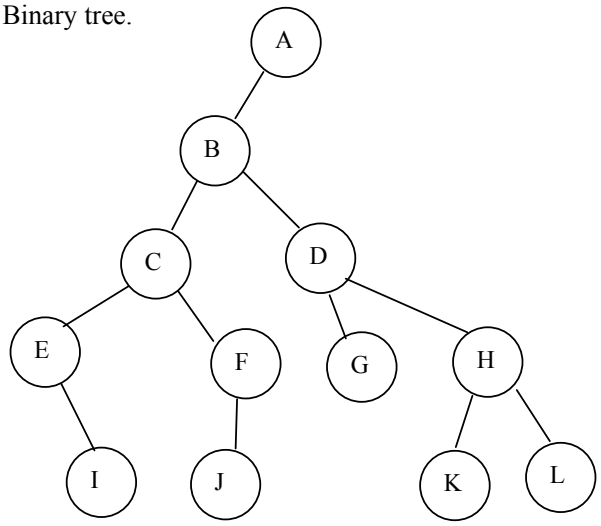
12) Consider the following algorithm.

(i) An empty node is created.
(ii) The node's information field is initialised to an integer e1.
(iii) The node is being included at the end of the list, and the next field is set to null.
(iv) The node is now included in the list by making the next field of the last node of the list a reference to the newly created node.
(v) The new node follows all the nodes of the list, but this fact has to be reflected in the value of the tail, which now becomes the reference to the new node.

Which of the following does the above algorithm describe?

(a) The process of adding a new node to the last node of the tree
(b) The process of adding a new node to any location of the list
(c) The process of adding a new node to the end of the list
(d) The process of deleting a node from the end of the list
(e) The process of deleting a node from the beginning of the list

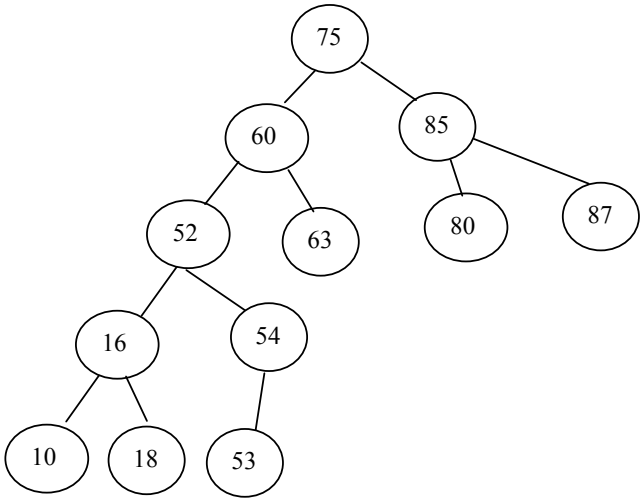
13) Consider the following Binary tree.



Which of the following expressions shows the pre-order, in-order, post-order respectively of the above tree?

(a) ABCEIFJDHGKL, EICFJBGDKHLA, IEJFCGKLHDBA
(b) ABCEIFJDGHKL, IEJFCGKLHDBA, EICFJBGDKHLA
(c) ABCEIFJDGHKL, EICFJBGDKHLA, IEJFCGKLHDBA
(d) EICFJBGDKHLA, ABCEIFJDGHKL, IEJFCGKLHDBA
(e) ABCEIFJDGHKL, EICFJBGDKHAL, IEJFCGKLHDBA

Question numbers 14-16 are based on the following Binary Search tree.



14) If one wants to insert a new node p=55, what is the correct insertion point without violating the binary search property?

(a) Right child of node number 18
(b) Right child of node number 53
(c) Left child of node number 63
(d) Right child of node number 54
(e) Left child of node number 54

15) If one wants to insert a new node p= 84 after inserting the new node p=55, to the above tree, what is the correct insertion point without violating the binary search property?

(a) Left child of node number 65
(b) Left child of node number 63
(c) Left child of node number 80
(d) Right child of node number 63
(e) Right child of node number 80

16) If one wants to delete node number 60 from the tree after inserting the new node p=55 and thereafter another at p=84, what is/are the suitable replacing node(s) without violating the binary search properties?

(a) Node number 63	(b) Node number 54	(c) Node number 52
(d) Node number 53	(e) Node number 85	

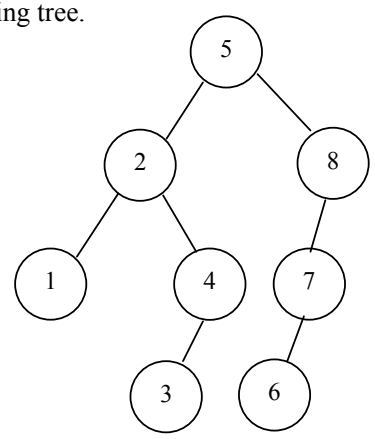
17) Consider the following incomplete Java program code segment intended for the insertion of new nodes to a Binary Search Tree.

```
public void insert (int e1) {
   IntBSTNode p = root,prev = null;
   while (  p != null) {
          prev = p;
           if (p.key < e1)
                 …….(i)……;
           else  …….(ii)……;
    }
if ( root = = null )
       root= new IntBSTNode(e1);
else if (prev.key <e1)
       ……(iii)…..  = new IntBSTNode(e1);
else   ……..(iv)….  = new IntBSTNode(e1);
}
```

What would be the correct way of filling the above blank positions?

(a) (i) p = p.leftt;	(ii) p= p. right;	(iii) prev.right	(iv) prev.left
(b) (i) p = p.right;	(ii) p= p.left;	(iii) prev.left	(iv) prev.right
(c) (i) p = p.right;	(ii) p= p.left;	(iii) prev.right	(iv) prev.left
(d) (i) p = p.next;	(ii) p= p.left;	(iii) prev.right	(iv) prev.left
(e) (i) p = p.right;	(ii) p= p.left;	(iii) prev.right	(iv) prev.p.left

18) Consider the following tree.



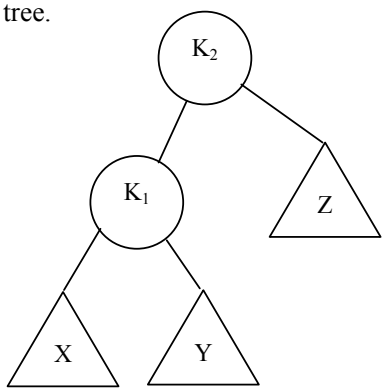
The above tree is a/an

(a) AVL tree.	(b) Binary Search Tree.	(c) Binary Tree.
(d) General Tree.	(e) Multiway search Tree.	

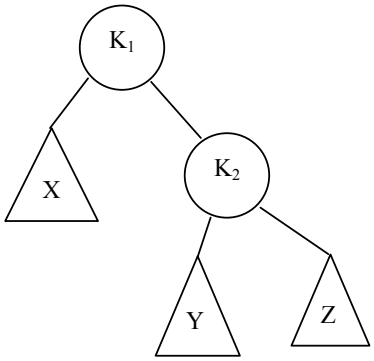
19) Which of the following statements is/are correct in connection with trees?

(a) It is an extension of a binary tree.
(b) It is a special kind of general tree.
(c) It is an extension of a Binary search tree.
(d) Balance factor is irrelevant for AVL trees.
(e) Parent key value is always greater than that of its children.

20) Consider the following tree.



The above tree is not an AVL tree. To convert it into AVL, a single rotation is needed. After such single rotation, the resultant tree is as follows.



For the above single rotation, the following Java program segment serves provided that the blanks are appropriately filled.

```
private static AvlNode  rotateWithLeftChild( AvlNode k2)
{
AvlNode k1=k2.left;
……..(i)…….
k1.right = k2;
……..(ii)…….= max( height ( k2.left),height(k2.right))+1;
k1.height= ……(iii)……( height ( k1.left),k2.height)+1;
return k1;
}
```

The correct way(s) for filling blank positions is/are

(a) (i) k2.left = k1.right; (ii) k2.height (iii) min	(b) (i) k1.left = k1.right; (ii) k2.height (iii) min
(c) (i) k2.left = k2.right; (ii) k2.height (iii) min	(d) (i) k2.left = k1.right; (ii) k2.height (iii) max
(e) (i) k2.left = k1.right; (ii) k2.right (iii) min	

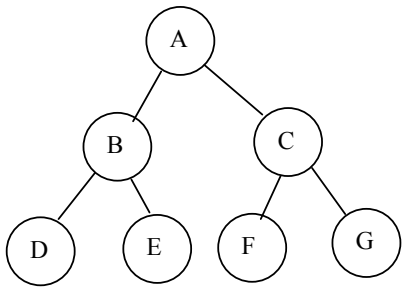
21) Consider the following paragraph with blank positions.

A/An. ……(i)…… tree of order n is a general tree in which each node has n or fewer sub trees and contains one fewer key than its ……(ii)…….. That is, if a node has four sub trees, it contains..(iii)……

Correct words for the blank positions are

(a) (i) multiway search	(ii) binary tree	(iii) three keys
(b) (i) Binary search	(ii) subtrees	(iii) three keys
(c) (i) multiway search	(ii) subtrees	(iii) three keys
(d) (i) multiway search	(ii) subtrees	(iii) five keys
(e) (i) AVL	(ii) subtrees	(iii) three keys

22) Consider the following AVL tree.



If one performs a left single rotation considering the Node A, what would be correct from among the following?

(a) New root is Node C.	(b) New root is Node B.
(c) Node A's right child is Node F.	(d) Node A's right child is Node C.
(e) Node G's left child and right child are null.	

23) Consider the following paragraph.

A tree is a data structure consisting of nodes connected to each other with references. Each node in a tree may be connected to one or two nodes. Each node contains one or more data fields and two references.

Which of the following is/are properly describing the above paragraph?

(a) Multiway search Trees	(b) Binary Trees	(c) Binary Search Trees
(d) General Trees	(e) AVL trees	

24) Consider the following four statements with blank positions.

If one finds a node with a/an ……. (i)……, he should stop the trace at that point.
Consider the node with imbalance and …….(ii)…….. on the layers immediately below this point on the path back to the new node.
If these three nodes lie in a ……(iii)…… apply a …….(iv)…… rotation.
If these three nodes lie in a dog-leg pattern (bend in the path), apply ……(v)……. rotation to correct the imbalance.

The above sentences are in connection with insertion of new nodes in to an AVL tree. Which of the following are correct words for the above blank positions?

(a) (i) balance	(ii)two nodes	(iii) straight line	(iv) single	(v) double
(b) (i) imbalance	(ii)three nodes	(iii) straight line	(iv) single	(v) double
(c) (i) imbalance	(ii)two nodes	(iii) dog-leg	(iv) single	(v) double
(d) (i) imbalance	(ii)two nodes	(iii) straight line	(iv) double	(v) single
(e) (i) imbalance	(ii)two nodes	(iii) straight line	(iv) single	(v) double

25) Which of the following statements is/are true in connection with graphs?

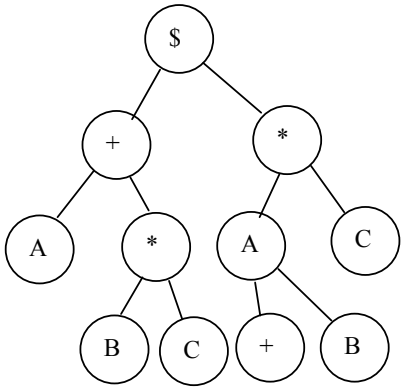
(a) If few nodes are associated with a Di-Graph, array based implementation is more useful than linked list implementation.
(b) If the graph contains n nodes, a total of n^2 locations must be used in array based implementation.
(c) If the graph is not weighted, the entire graph can de described using the Adjacency matrix.
(d) Warshall's Algorithm increases the efficiency of finding the transitive closure.
(e) An example of a real life situation that can be modelled by a graph is the Air path System.

26) Consider the following expression.

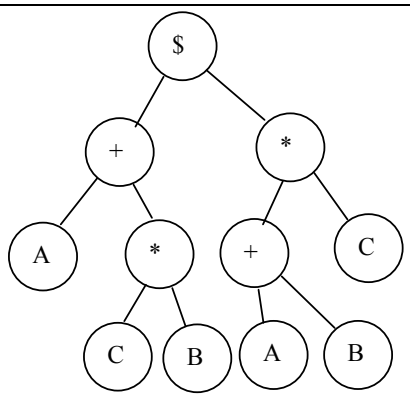
(A+B*C)$((A+B)*C)

Which of the following trees show(s) the equivalent expression tree for the above expression?
Note: $ denotes power.

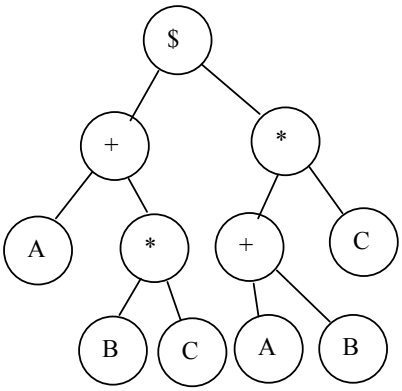
(a)



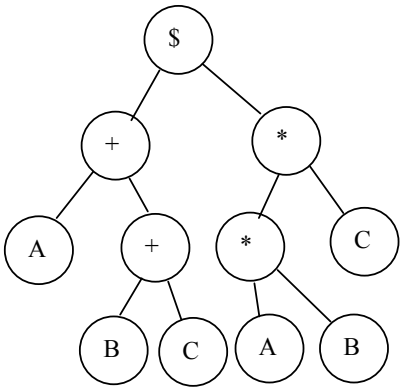
(b)



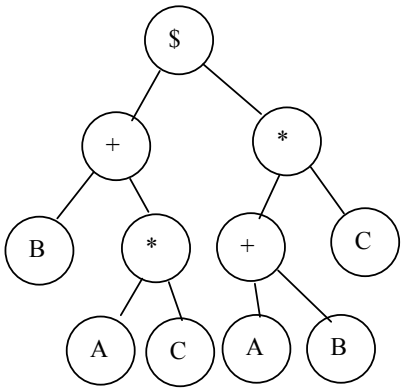
(c)



(d)



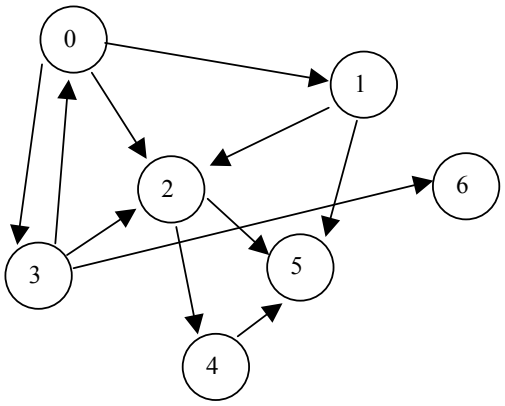
(e)



27) If one chooses the middle key as the pivot, what would be the Big-O value for the Best case of the Quick Sort Algorithm?

(a) n^2	(b) n	(c) a constant
(d) n^3	(e) n log n	

Note: Question numbers 28-30 are based on the following directed graph.



28) Which of the following edge table representations describe(s) the above graph?

(a)

0	1 2 3
1	2 5
2	4 5
3	0 2 6
4	5
5	
6	

(b)

0	1 2 3 3
1	0 2 3
2	0 1 3 4 5
3	0 0 2
4	2 5
5	1 2 4
6	3

(c)

0	1 2 3
1	0 2 3
2	0 1 3 4 5
3	0 2
4	2 5
5	1 2 4
6	3

(d)

0	1 2 3
1	0 2 3
2	0 1 3 4 5
3	0 0 2
4	2 5
5	1 2 4
6	3

(e)

0	1 2 3 3
1	0 2 3
2	0 1 3 4 5
3	0 0 2
4	2 5
5	
6	

29) If one chooses the starting vertex as Number 0, which of the following is/are the order of output nodes in the breadth first traversal?

(a) 0 1 2 3 5 4 6	(b) 0 2 3 1 4 5 6	(c) 0 2 1 3 4 5 6
(d) 0 1 2 4 5 6 3	(e) 0 1 2 5 6 4 3	

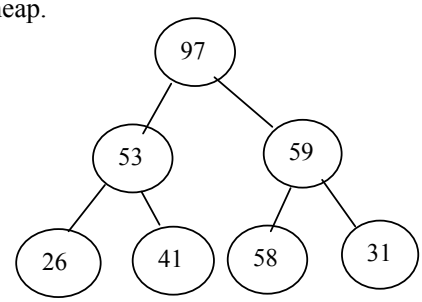
30) If one chooses the starting vertex as No 0, which of the following is/are the order of output nodes in the depth first traversal?

(a) 0 2 1 3 4 5 6	(b) 0 1 2 4 5 3 6	(c) 1 2 4 5 0 3 6
(d) 0 2 4 5 3 6 1	(e) 0 1 2 3 5 4 6	

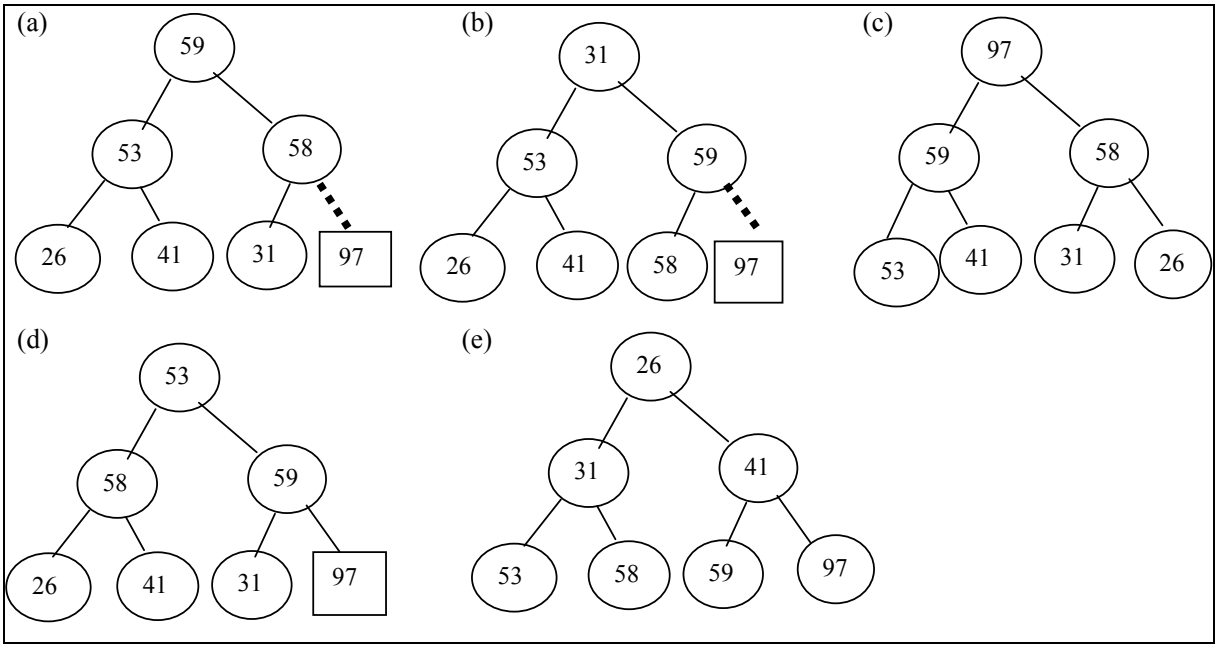
31) Which of the following statements is/are correct in connection with Graphs?

(a) There are three kinds of Graphs, namely Directed Graphs, Un-directed Graphs and Digraphs.
(b) The In _degree of a node n is the number of arcs which have n as the head.
(c) The path matrix of any Di-graph can be calculated as $path_n[I][J]=adj_1[I][J]$ or $adj_2[I][J]$ or $adj_3[I][J]$ or …… $adj_n[I][J]$
(d) Transitive closure can be calculated using the Path Matrix.
(e) Transitive closure can be calculated using the Adjacency Matrix.

32) Consider the following heap.



What would be the resulting heap, after performing the DeleteMax operation (delete maximum node from the heap) without violating the heap conditions?



33) Consider the following selection sort algorithm and the integer data set given thereafter.

For a list with L elements numbered from 0 to L-1, the selection sort algorithms is :

1. Set marker to L-1 (beginning with the last element in the list)
2. While marker >0 ;
   - i. Find the largest element in the range numbered from 0 to marker.
   - ii. Swap the largest element with the element at the location marker.
   - iii. Increase marker back by 1

Data set: { 4 10 2 8 3 }

If the above algorithm works on the given data set , what would be the pair of swapping values during the process of sorting?

(a) (3,10)	(b) (4,2)	(c) (2,3)
(d) (10,2)	(e) (8,4)	

34) This question also is based on the algorithm given in question 33. If one wants to convert Line No (ii) of the algorithm into Java Code, what would be the more appropriate coding?

(a)
```
public  static swap(int [ ] a,int i ,int j){
    int temp;
    temp= a[i];
    a[i]= a[j];
    a[j] = temp;
    }
```

(b)
```
public  static swap(int [ ] a,int i ,int j){
    int temp;
    temp= a[j];
    a[j]=a[i];
    a[i] = temp;
    }
```

(c)
```
public void swap( ){
    int temp,a,b;
    a=temp;
    b=a;
    b = temp;
    }
```

(d)
```
public void swap( ){
    int temp,a,b,c
    a=temp;
    b=temp
    c= temp;
    }
```

(e)
```
public void swap( ){
    int temp,a,b;
    temp=a;
    b=a;
    b = temp;
    }
```

35) Which of the following statements is/are correct in connection with the binary search algorithm?

(a) The most efficient method of searching a sequential table without the use of auxiliary indices or tables is the binary search algorithm.
(b) Basically, the argument is compared with the key of the middle element of the table. If they are equal, the search is successful; otherwise, either the upper half or the lower half of the table must be searched in a similar manner.
(c) Any kind of data set can be used.
(d) Time complexity of the best case is O (n^2).
(e) Time complexity is O (n log n).

36) Consider the following program segment.

```
Public int pqr( int [ ] data, int x, int l, int h){
        int mid
        if (l>h) {
        return -1;
        } else {
        mid= (l+h)/2;
        if ( data[mid] = = x {
        return mid;
        } else {
        if( x< data[mid]) return pqr(data,x,l,mid-1);
        else return pqr(data,x,mid+1,h);
        }
        }
        }
```

What is the above Java program segment intended to do?

(a) Searching for element x from any set of data using the sequential search algorithm
(b) Searching for element x from any set of data using the binary search algorithm
(c) Searching for element x from the sorted set of data using the sequential search algorithm
(d) Searching for element x from the sorted set of data using the binary search algorithm
(e) Finding the middle element from any given set of data

37) Consider the following algorithm with the given steps.

1. Choose the source vertex
2. Define a set S of vertices and initialise it to contain only the source vertex. The set S will store those vertices to which a shortest distance has been found as the algorithm progresses.
3. Label each vertex not in the set S with the minimum distance from the source vertex, but only if this vertex can be connected by a single edge with the source. If there is no single edge connection between the source and the vertex not in S, label the vertex not in S with a distance of infinity.
4. Add to S that vertex with the smallest distance as determined in step 3 (or step 5 or any subsequent iteration of the algorithm)
5. Check the distances to all vertices not in S. If the latest addition to S in step 4 provides a single edge connection between a vertex in S and the vertices not in S, or a smaller path exists, then adjust the distance to the vertex. Otherwise, do not change the distance.
6. Repeat the algorithm from step 4 until either all vertices are in the set S or all vertices not in S have an infinite distance to them. In the latter case, these vertices are not reachable from the source vertex at any cost.

Which of the following describe(s) the above algorithm?

(a) Depth First Search Algorithm	(b) Breadth First Search Algorithm
(c) Shortest Path Algorithm	(d) Warshall's Algorithm
(e) Transitive Closure Algorithm	

38) Consider the following Java program segment and the data set given thereafter.

```
public static void insertionSort ( Comparable [ ] a)
{
 int j;
 for ( int p = 1; p < a.length; p++)
{
   Comparable tmp =a[p];
   for (j =p; j>0 && tmp.compareTo ( a[j-1]) < 0; j--)
       a[j] = a[j-1];
   a[j] =tmp;
  }
}
```

Data set: {34 8 64 51 32 21}

If the above Java program segment works on the given data set, which of the following give(s) the number of positions moved during the process of execution, at each pass?

(a) 1 0 1 3 4 5	(b) 0 1 2 3 4 5	(c) 1 0 1 3 4
(d) 0 1 1 3 5 2	(e) 1 0 1 2 4	

39) {21, 48, 37, 54, 45, 33, 76, 91, 09, 78} is a given set of integers. If one uses straight merge sort to sort the above set of integers, what would be the intermediate output file(s) during the 1st pass?

(a) 09, 21, 33, 37,45, 48, 54,76,78, 91	(b) 91,78, 76, 54, 48, 45, 37, 33, 21, 09
(c) 21, 48, 37, 54, 33, 45, 76, 91,78, 09	(d) 21, 48, 37, 54, 33, 45, 76, 91, 09, 78
(e) 48, 21, 54, 37, 45, 33, 91, 76, 78, 09	

40) Suppose T_1 (N)=O (f (N)) and T_2 (N)=O (g(N)). Which of the following is/are true?

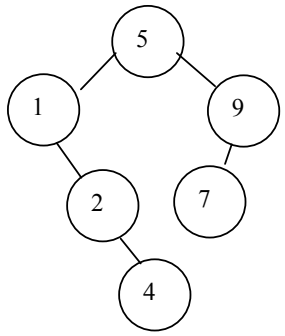
(a) T_1 (N)+ T_2 (N)=O (f (N))	(b) T_1 (N)- T_2 (N)=O (f (N))	(c) $T_1(N)/T_2(N)=O(1)$
(d) T_1 (N)=O (T_2 (N))	(e) T_1 (N)+ T_2 (N)=max(O (f (N)), O (g(N)))	

41) Which of the following is/are correct in connection with the time complexity of the quick sort algorithm in the worst case, best case and average case respectively?

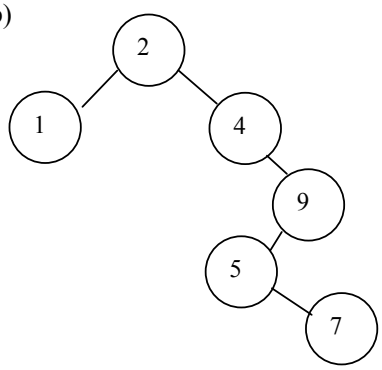
(a) n^2, n, nlog n	(b) nlog n, nlog n, nlog n	(c) n^2, nlog n, nlog n
(d) n^2, nlog n, n^2	(e) n, n, nlog n	

42) Consider the result of inserting 3,1,4,9,2,5,7 into an empty Binary search tree. What would be the resulting tree after deleting the root?

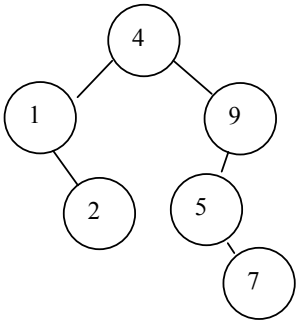
(a)



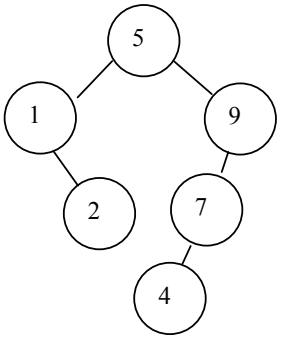
(b)



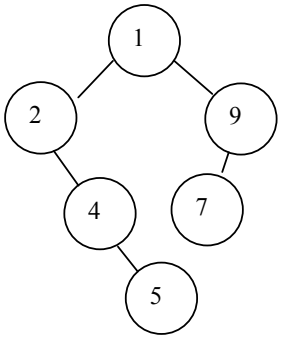
(c)



(d)



(e)



43) The following shows a series of stack operations.

(i) Push(5)
(ii) Push(8)
(iii) Isempty( )
(iv) Pop( )
(v) Push(3)
(vi) Pop( )
(vii) Pop( )
(viii) Pop( )
(ix) Isfull( )
(x) Push(3)

If the above series of operations is performed, what would be the set returned values?

(a) {false, 8, 3, 5, error, false, 3}	(b) {5, 8, false, 8, 3, 5, error, false, 3}
(c) {false, 8, 3, 5, error, false}	(d) {false, 8,3, 5, false}
(e) {false, 8, 3, 5, error, true}	

44) Consider the following operations and their definitions.

(i) Clear( ) :– Clear the stack.
(ii) Is_Empty( ) :– Remove all items from the stack.
(iii) Insert(x) :- Insert element x in to any location of the stack.
(iv) Delete(p) :- Delete the element from the position p.
(v) Top( ) :- Return and remove the topmost element from the stack.

Which of the above operations is/are valid in stacks according to the basic definitions?

(a) (i), (ii) and (v) only	(b) (i) and (v) only
(c) (i) only	(d) (i) and (ii) only
(e) (i), (ii), (iii), (iv) and (v)	

45 Consider the following rules.

Rule 1: **For Loop**: The running time of a **for** loop is at most the running time of the statements inside the **for** loop (including test) times the number of iterations

Rule 2 : **Nested Loop**: The total running time of a statement inside a group of nested loops is the running time of the statement multiplied by the product of the sizes of all the loops.

Rule 3 : **IF/ELSE** : For the fragment

```
if( condition)
   S1
else
   S2
```

the running time of an if/else statement is never more than the running time of the test plus the larger of the two numbers namely, the running time of S1 and that of S2.

Which of the following is a / are valid rule(s) in connection with analysis of algorithms?

(a) Rule 3 only	(b) Rule 1 and Rule 3 only	(c) Rule 2 and Rule 3 only
(d) Rule 2 only	(e) All	

*****