MOCK TEST 2 DS

Total points 20/20

Email *

rkg16697@gmail.com

Enter Roll No: *

1168

Enter Name: *

Rishikesh Gupta

✓ 1)The Infix equivalent of the prefix * + ab – cd is _____. 1/1

- (a) (a+b) * (c−d) ✓
- (b) (a+b) − (c*d)
- (c) (a*b)−(c+d)
- (d) (a−b)*(c+d)

✓ 2)The postfix equivalent of the prefix * + ab – cd is_____. 1/1

- (a) ab + cd – * ✓
- (b) abcd + – *
- (c) ab + cd * –
- (d) ab + – cd *

✓ 3)The postfix equivalent of the infix expression a+b+c+d is _____. 1/1

- (a) abcd+++
- (b) ab+c+d+ ✓
- (c) ab+cd++
- (d) (a–b)*(c+d)

✓ 4)The prefix equivalent of the infix expression a+b+c+d is _____. 1/1

- (a) +ab+c+d
- (b) +++abcd ✓
- (c) ++ab+cd+
- (d) abcd++++

✓ 5)The postfix equivalent of the infix expression a+b/c*d-e/f is _____. 1/1

- (a) ab+cd*/ef-/
- (b) abcd*+/ef-/
- (c) ab+cd*/ef/-
- (d) abc/d*+ef/- ✓

✓ 6)The prefix equivalent of the infix expression a+b/c*d-e/f is _____. 1/1

- (a) +abc-*/ef
- (b) +/*-/abcdef ✓
- (c) -+a*/bcd/ef
- (d) +a*/bcd-/ef

✓ 7)The postfix equivalent of the infix expression a+b/c-d*e-f is _____. 1/1

- (a) abc/+de*-f- ✓
- (b) abcd*+/ef-/
- (c) ab+cd*/ef/-
- (d) abc/d*+ef/-

✓ 8)The prefix equivalent of the infix expression a+b/c-d*e-f is _____. 1/1

- (a) +abc-*/ef
- (b) --+a/bc*def ✓
- (c) -+a*/bcd/ef
- (d) +a*/bcd-/ef

✓ 9) The infix equivalent of the postfix ab+cd+ef*-/ is_____. 1/1

- (a) ((a+b)/((c+d))-(e*f)) ✓
- (b) (a+b) - (c+d)/(e*f)
- (c) (a+b)*(c+d)-(e/f)
- (d) ((a+b)/(c+d))-(e*f)

✓ 10) The infix equivalent of the postfix ab*cd/+e- is _____. 1/1

- (a) a+b*c/d-e
- (b) a*b+c/d-e ✓
- (c) (a*b)-(c/d)+e
- (d) a*b-c/d+e

✓ 11)The prefix equivalent of the postfix ab*cd/+e- is _____. 1/1

- (a) +-/abc*de
- (b) -+*abc/de
- (c) -+*ab/cde ✓
- (d) *ab/+cd-e

✓ 12)Pick the correct prefix form to the given infix expression: {a*[b/(c-d)*f]/g}/[e+h] 1/1

- (a) //*a/b*-cdfg+ch
- (b) abcd-f*/g/*eh+/ ✓
- (c) //*a*/b-cdfg+eh
- (d) //*ab*/-cdfg+eh

✓ 13)For the expression ((A + B) * C – (D – E)/(F + G)), the equivalent Postfix notation is 1/1

- (a) AB + C * DE - - / FG +
- (b) AB + C * DE - FG + /- ✓
- (c) AB + C * DE - - FG + /
- (d) AB + C - DE - * FG + /

✓ 14)What is the postfix expression for the corresponding infix expression? 1/1
a+b*c+(d*e)

- (●) abc*+de* ✓
- ( ) abc+*de*
- ( ) a+bc*de+*
- ( ) abc*+(de)*

✓ 15)What would be the solution to the given prefix notation? - + 5 / 10 5 5 1/1

- (●) 2 ✓
- ( ) 5
- ( ) 10
- ( ) 7

✓ 16)What would be the solution to the given prefix notation? / / / / 16 4 2 1 1/1

- (●) 1 ✓
- ( ) 4
- ( ) 2
- ( ) 8

✓ 17)The equivalent infix expression and value for the postfix form 1 2 + 3 * 4 5 * - will be 1/1

- 1 + 2 * 3 - 4 * 5 and -13
- (2 + 1) * (3 - 4) * 5 and 13
- 1 + 2 * (3 - 4) * 5 and -11
- (1 + 2) * 3 - (4 * 5) and -11 ✓

✓ 18)What is the value of the postfix expression 2 3 + 4 5 6 - - * 1/1

- 19
- 21 ✓
- 23
- -2

✓ 19) What would be the Prefix notation for the given equation? a+b-c/d&e|f 1/1

- A) |&-+ab/cdef ✓
- B) &|-+ab/cdef
- C) |&-ab+/cdef
- D) |&-+/abcdef

✓ 20)What would be the Prefix notation and Postfix notation for the given equation? A+B+C 1/1

- (●) A) ++ABC and AB+C+ ✓
- ( ) B) AB+C+ and ++ABC
- ( ) C) ABC++ and AB+C+
- ( ) D) ABC+ and ABC+

ADS_Test1

Total points 20/20

Email *

rkg16697@gmail.com

RollNo *

1168

✓ 1)Elements can be retrieved by index in .....? * 1/1

- ○ linked lists
- ◉ linear arrays ✓
- ○ both of above
- ○ none of above

✓ 2) Efficiency of an algorithm is measured by * 1/1

- ○ Time and Capacity complexity
- ◉ Time and Space complexity ✓
- ○ Speed and Space complexity
- ○ Speed and Capacity complexity

✓ 3) State True or False about array and linked list I. Retrieval of element will be faster in array than link list? II. Search operation in both array and link list are leaner. * 1/1

- ○ True, False
- ○ False, True
- ○ False, False
- ◉ True, True ✓

✓ 4) True statements about Stack and Queue are I. Stack and Queue both are linear data structures II. Stack is non- linear data structure. III. Stack is LIFO IV. Queue is FIFO * 1/1

- ○ I and II only
- ○ I and III
- ◉ I, III and IV ✓
- ○ All are correct.

✓ 5) If the elements “A”, “B”, “C” and “D” are placed in a queue and are deleted one at a time, in what order will they be removed? * 1/1

- ◉ ABCD ✓
- ○ DCBA
- ○ DCAB
- ○ ABDC

✓ 6) A data structure in which elements can be inserted or deleted at/from both ends but not in the middle is? * 1/1

- ○ Queue
- ○ Circular queue
- ◉ Dequeue ✓
- ○ Priority queue

✓ 7) A normal queue, if implemented using an array of size MAX_SIZE, gets full when? * 1/1

- ◉ Rear = MAX_SIZE − 1 ✓
- ○ Front = (rear + 1)mod MAX_SIZE
- ○ Front = rear + 1
- ○ Rear = front

✓ 8) Queues serve major role in _____ * 1/1

- ○ Simulation of recursion
- ○ Simulation of arbitrary linked list
- ◉ Simulation of limited resource allocation ✓
- ○ Simulation of heap sort

✓ 9)Which of the following is not the type of queue? * 1/1

- ○ Ordinary queue
- ◉ Single ended queue ✓
- ○ Circular queue
- ○ Priority queue

✓ 10) Process of inserting an element in stack is called _____ * 1/1

- ○ Create
- ◉ Push ✓
- ○ Evaluation
- ○ Pop

✓ 11) In a stack, if a user tries to remove an element from an empty stack it is called _____ * 1/1

- ◉ Underflow ✓
- ○ Empty collection
- ○ Overflow
- ○ Garbage Collection

✓ 12) Entries in a stack are “ordered”. What is the meaning of this statement? * 1/1

- ○ A collection of stacks is sortable
- ○ Stack entries may be compared with the ‘<‘ operation
- ○ The entries are stored in a linked list
- ◉ There is a Sequential entry that is one by one ✓

✓ 13) The data structure required to check whether an expression contains a balanced parenthesis is? * 1/1

- ◉ Stack ✓
- ○ Queue
- ○ Array
- ○ Tree

✓ 14) What data structure would you mostly likely see in non recursive implementation of a recursive algorithm? * 1/1

- ○ Linked List
- ◉ Stack ✓
- ○ Queue
- ○ Tree

✓ 15) The process of accessing data stored in a serial access memory is similar to manipulating data on a _____ * 1/1

- ○ Heap
- ○ Binary Tree
- ○ Array
- ◉ Stack ✓

✓ 16) Which data structure is used for implementing recursion? * 1/1

- ○ Queue
- ◉ Stack ✓
- ○ Array
- ○ List

✓ 17) Which of the following statement(s) about stack data structure is/are NOT correct? * 1/1

- ○ Linked List are used for implementing Stacks
- ○ Top of the Stack always contain the new node
- ◉ Stack is the FIFO data structure ✓
- ○ Null link is present in the last node at the bottom of the stack

✓ 18) Consider the following operation performed on a stack of size 5. Push(1); Pop(); Push(2); Push(3); Pop(); Push(4); Pop(); Pop(); Push(5); * 1/1

- (●) 1 ✓
- ( ) 2
- ( ) 3
- ( ) 4

✓ 19) Which of the following is not an inherent application of stack? * 1/1

- ( ) Reversing a string
- ( ) Evaluation of postfix expression
- ( ) Implementation of recursion
- (●) Job scheduling ✓

✓ 20) Which of the following real world scenarios would you associate with a stack data structure? * 1/1

- (●) piling up of chairs one above the other ✓
- ( ) people standing in a line to be serviced at a counter
- ( ) offer services based on the priority of the customer
- ( ) tatkal Ticket Booking in IRCTC

DS Using Java

Total points 18/20

Email *

rkg16697@gmail.com

Enter your Name: *

Rishikesh Gupta

Enter Roll No: *

1168

✓ 1. Which of the following is not a disadvantage to the usage of array? * 1/1

- a) Fixed size
- b) There are chances of wastage of memory space if elements inserted in an array are lesser than the allocated size
- c) Insertion based on position
- d) Accessing elements at specified positions ✓

✓ 2.What is the functionality of the following code? * 1/1

```
public void function(Node node)
{
        if(size == 0)
                head = node;
        else
        {
                Node temp,cur;
                for(cur = head; (temp = cur.getNext())!=null; cur = temp);
                cur.setNext(node);
        }
        size++;
}
```

- a) Inserting a node at the beginning of the list
- b) Deleting a node at the beginning of the list
- c) Inserting a node at the end of the list ✓
- d) Deleting a node at the end of the list

✓ 3. Which of these is not an application of a linked list? * 1/1

- a) To implement file systems
- b) For separate chaining in hash-tables
- c) To implement non-binary trees
- d) Random Access of elements ✓

✗ 4.What is the functionality of the following piece of code? * 0/1

```
public int function(int data)
{
        Node temp = head;
        int var = 0;
        while(temp != null)
        {
                if(temp.getData() == data)
                {
                        return var;
                }
                var = var+1;
                temp = temp.getNext();
        }
        return Integer.MIN_VALUE;
}
```

- a) Find and delete a given element in the list
- b) Find and return the given element in the list ✗
- c) Find and return the position of the given element in the list
- d) Find and insert a new element in the list

Correct answer

- c) Find and return the position of the given element in the list

✓ 5.What kind of linked list is best to answer questions like “What is the item at position n?” * 1/1

- a) Singly linked list
- b) Doubly linked list
- c) Circular linked list
- (●) d) Array implementation of linked list ✓

✓ 6. Linked list data structure offers considerable saving in _____ * 1/1

- a) Computational Time
- b) Space Utilization
- (●) c) Space Utilization and Computational Time ✓
- d) Speed Utilization

✓ 7.Which of the following is false about a doubly linked list? * 1/1

- a) We can navigate in both the directions
- b) It requires more space than a singly linked list
- c) The insertion and deletion of a node take a bit longer
- (●) d) Implementing a doubly linked list is easier than singly linked list ✓

✓ 8.How do you calculate the pointer difference in a memory efficient double linked list? * 1/1

- a) head xor tail
- b) pointer to previous node xor pointer to next node ✓
- c) pointer to previous node – pointer to next node
- d) pointer to next node – pointer to previous node

✓ 9. Consider the following doubly linked list: head-1-2-3-4-5-tail. What will be the list after performing the given sequence of operations? * 1/1

```
Node temp = new Node(6,head,head.getNext());
Node temp1 = new Node(0,tail.getPrev(),tail);
head.setNext(temp);
temp.getNext().setPrev(temp);
tail.setPrev(temp1);
temp1.getPrev().setNext(temp1);
```

- a) head-0-1-2-3-4-5-6-tail
- b) head-1-2-3-4-5-6-tail
- c) head-6-1-2-3-4-5-0-tail ✓
- d) head-0-1-2-3-4-5-tail

✓ 10.What is the functionality of the following piece of code? Select the most appropriate. * 1/1

```
public void function(int data)
{
        int flag = 0;
        if( head != null)
        {
                Node temp = head.getNext();
                while((temp != head) && (!(temp.getItem() == data)))
                {
                        temp = temp.getNext();
                        flag = 1;
                        break;
                }
        }
        if(flag)
                System.out.println("success");
        else
                System.out.println("fail");
}
```

- a) Print success if a particular element is not found
- b) Print fail if a particular element is not found ✓
- c) Print success if a particular element is equal to 1
- d) Print fail if the list is empty

✓ 11.Which of the following application makes use of a circular linked list? * 1/1

- a) Undo operation in a text editor
- b) Recursive function calls
- c) Allocating CPU to resources ✓
- d) Implement Hash Tables

✗ 12.What is the functionality of the following code? Choose the most appropriate answer. * 0/1

```
public int function()
{
        if(head == null)
                return Integer.MIN_VALUE;
        int var;
        Node temp = head;
        Node cur;
        while(temp.getNext() != head)
        {
                cur = temp;
                temp = temp.getNext();
        }
        if(temp == head)
        {
                var = head.getItem();
                head = null;
                return var;
        }
        var = temp.getItem();
        cur.setNext(head);
        return var;
}
```

- ○ a) Return data from the end of the list
- ○ b) Returns the data and deletes the node at the end of the list
- ◉ c) Returns the data from the beginning of the list ✗
- ○ d) Returns the data and deletes the node from the beginning of the list

Correct answer

- ◉ b) Returns the data and deletes the node at the end of the list

✓ 13. In linked list implementation of a queue, where does a new element be inserted? * 1/1

- a) At the head of link list
- b) At the centre position in the link list
- c) At the tail of the link list ✓
- d) At any position in the linked list

✓ 14.In linked list implementation of a queue, front and rear pointers are tracked. Which of these pointers will change during an insertion into EMPTY queue? * 1/1

- a) Only front pointer
- b) Only rear pointer
- c) Both front and rear pointer ✓
- d) No pointer will be changed

✓ 15.In linked list implementation of a queue, the important condition for a queue to be empty is? * 1/1

- a) FRONT is null ✓
- b) REAR is null
- c) LINK is empty
- d) FRONT==REAR-1

✓ 16.Which of the following is true about linked list implementation of queue? * 1/1

- (●) a) In push operation, if new nodes are inserted at the beginning of linked list, then in pop operation, nodes must be removed from end ✓
- ( ) b) In push operation, if new nodes are inserted at the beginning, then in pop operation, nodes must be removed from the beginning
- ( ) c) In push operation, if new nodes are inserted at the end, then in pop operation, nodes must be removed from end
- ( ) d) In push operation, if new nodes are inserted at the end, then in pop operation, nodes must be removed from beginning

✓ 17. What is the number of moves required to solve Tower of Hanoi problem for k disks? * 1/1

- ( ) a) 2k − 1
- ( ) b) 2k + 1
- ( ) c) 2k + 1
- (●) d) 2k − 1 ✓

✓ 18. Which application of stack is used to ensure that the pair of parentheses is properly nested? * 1/1

- (●) a) Balancing symbols ✓
- ( ) b) Reversing a stack
- ( ) c) Conversion of an infix to postfix expression
- ( ) d) Conversion of an infix to prefix expression

✓ 19. Which of the following statement is incorrect with respect to balancing symbols algorithm? * 1/1

- a) {[()]}
- b) ([ )] ✓
- c) {( )}
- d) { [ ] }

✓ 20.Parentheses are simply ignored in the conversion of infix to postfix expression. * 1/1

- a) True
- b) False ✓

DS Using Java

Total points 20/20

Email *

rkg16697@gmail.com

Enter Your Name: *

Rishikesh Gupta

Enter Roll No: *

1168

✓ 1. How many children does a binary tree have? * 1/1

- ○ a) 2
- ○ b) any number of children
- ◉ c) 0 or 1 or 2 ✓
- ○ d) 0 or 1

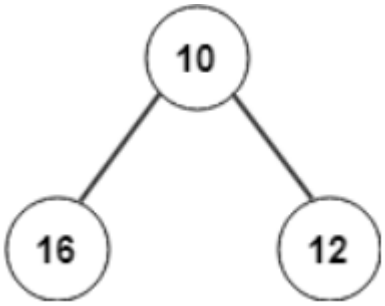
✓ 2.Consider a situation of writing a binary tree into a file with memory storage efficiency in mind, is array representation of tree is good? * 1/1

- ( ) a) yes because we are overcoming the need of pointers and so space efficiency
- ( ) b) yes because array values are indexable
- (•) c) No it is not efficient in case of sparse trees and remaning cases it is fine ✓
- ( ) d) No linked list representation of tree is only fine

✓ 3.Can a tree stored in an array using either one of inorder or post order or pre order traversals be again reformed? * 1/1

- ( ) a) Yes just traverse through the array and form the tree
- (•) b) No we need one more traversal to form a tree ✓
- ( ) c) No in case of sparse trees
- ( ) d) Yes by using both inorder and array elements

✓ 4.The following given tree is an example for? * 1/1



- (●) a) Binary tree ✓
- ( ) b) Binary search tree
- ( ) c) Fibonacci tree
- ( ) d) AVL tree

✓ 5. A binary tree is a rooted tree but not an ordered tree. * 1/1

- ( ) a) true
- (●) b) false ✓

✓ 6.How many common operations are performed in a binary tree? * 1/1

- ( ) a) 1
- ( ) b) 2
- (●) c) 3 ✓
- ( ) d) 4

✓ 7.What is the traversal strategy used in the binary tree? * 1/1

- ○ a) depth-first traversal
- ◉ b) breadth-first traversal ✓
- ○ c) random traversal
- ○ d) Priority traversal

✓ 8.If binary trees are represented in arrays, what formula can be used to locate a left child, if the node has an index i? * 1/1

- ◉ a) 2i+1 ✓
- ○ b) 2i+2
- ○ c) 2i
- ○ d) 4i

✓ 9. What does the following piece of code do? * 1/1

```
public void func(Tree root)
{
        func(root.left());
        func(root.right());
        System.out.println(root.data());
}
```

- a) Preorder traversal
- b) Inorder traversal
- c) Postorder traversal ✓
- d) Level order traversal

✓ 10.What is the speciality about the inorder traversal of a binary search tree? * 1/1

- a) It traverses in a non increasing order
- b) It traverses in an increasing order ✓
- c) It traverses in a random fashion
- d) It traverses based on priority of the node

✓ 11.The number of edges from the root to the node is called _____ of the tree. * 1/1

- ○ a) Height
- ◉ b) Depth ✓
- ○ c) Length
- ○ d) Width

✓ 12.What is a full binary tree? * 1/1

- ◉ a) Each node has exactly zero or two children ✓
- ○ b) Each node has exactly two children
- ○ c) All the leaves are at the same level
- ○ d) Each node has exactly one or two children

✓ 13. Which of the following is incorrect with respect to binary trees? * 1/1

- ○ a) Let T be a binary tree. For every k ≥ 0, there are no more than 2k nodes in level k
- ○ b) Let T be a binary tree with λ levels. Then T has no more than 2λ − 1 nodes
- ○ c) Let T be a binary tree with N nodes. Then the number of levels is at least ceil(log (N + 1))
- ◉ d) Let T be a binary tree with N nodes. Then the number of levels is at least floor(log (N + 1)) ✓

✓ 14. Advantages of linked list representation of binary trees over arrays? * 1/1

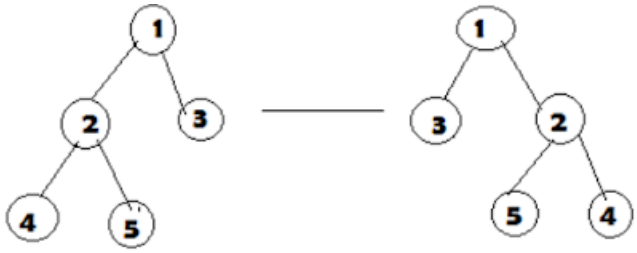
- a) dynamic size
- b) ease of insertion/deletion
- c) ease in randomly accessing a node
- (●) d) both dynamic size and ease in insertion/deletion ✓

✓ 15.What is missing in this logic of finding a path in the tree for a given sum (i.e checking whether there will be a path from roots to leaf nodes with given sum)? * 1/1

```
checkSum(struct bin-treenode *root , int sum) :
  if(root==null)
    return sum as 0
  else :
     leftover_sum=sum-root_node-->value
     //missing
```

- (●) a) code for having recursive calls to either only left tree or right trees or to both subtrees depending on their existence ✓
- b) code for having recursive calls to either only left tree or right trees
- c) code for having recursive calls to either only left tree
- d) code for having recursive calls to either only right trees

✓ 16. What must be the missing logic below so as to print mirror of a tree as below as an example? * 1/1




```
if(rootnode):
  mirror(rootnode-->left)
  mirror(rootnode-->right)

  //missing

end
```

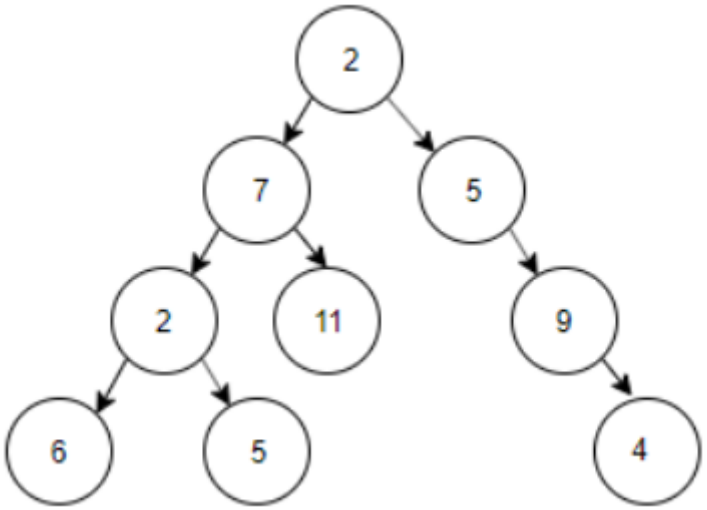
- (●) a) swapping of left and right nodes is missing ✓
- ( ) b) swapping of left with root nodes is missing
- ( ) c) swapping of right with root nodes is missing
- ( ) d) nothing is missing

✓ 17. What is the code below trying to print? * 1/1

```
void print(tree *root,tree *node)
{
  if(root ==null) return 0
  if(root-->left==node || root-->right==node) || print(root->left,node)
  ||printf(root->right,node)
  {
     print(root->data)
  }
}
```

- a) just printing all nodes
- b) not a valid logic to do any task
- c) printing ancestors of a node passed as argument ✓
- d) printing nodes from leaf node to a node passed as argument

✓ 18. For the tree below, write the pre-order traversal. * 1/1



- (●) a) 2, 7, 2, 6, 5, 11, 5, 9, 4 ✓
- ( ) b) 2, 7, 5, 2, 6, 9, 5, 11, 4
- ( ) c) 2, 5, 11, 6, 7, 4, 9, 5, 2
- ( ) d) 2, 7, 5, 6, 11, 2, 5, 4, 9

✓ 19.Consider the following data. The pre order traversal of a binary tree is A, B, E, C, D. The in order traversal of the same binary tree is B, E, A, D, C. The level order sequence for the binary tree is _____ * 1/1

- ( ) a) A, C, D, B, E
- (●) b) A, B, C, D, E ✓
- ( ) c) A, B, C, E, D
- ( ) d) D, B, E, A, C

✓ 20. Which of the following graph traversals closely imitates level order traversal of a binary tree? * 1/1

- ○ a) Depth First Search
- ◉ b) Breadth First Search ✓
- ○ c) Depth & Breadth First Search
- ○ d) Binary Search

DS_Test1

Total points 15/20

Email *

rkg16697@gmail.com

Name *

Rishikesh gupta

Roll No. *

1168

✓ 1) Which of the following is the advantage of the array data structure? 1/1

- ○ Elements of mixed data types can be stored.
- ◉ Easier to access the elements in an array ✓
- ○ Index of the first element starts from 1.
- ○ Elements of an array cannot be sorted

✓ 2) Which of the following is the disadvantage of the array? 1/1

- ○ Stack and Queue data structures can be implemented through an array.
- ○ Index of the first element in an array can be negative
- ◉ Wastage of memory if the elements inserted in an array are lesser than the allocated size ✓
- ○ Elements can be accessed sequentially.

✗ 3) Which one of the following is not the application of the stack data structure 0/1

- ○ String reversal
- ◉ Recursion ✗
- ○ Backtracking
- ○ Asynchronous data transfer

Correct answer

- ◉ Asynchronous data transfer

✓ 4) Which of the following is not the correct statement for a stack data structure? 1/1

- ○ Arrays can be used to implement the stack
- ◉ Stack follows FIFO ✓
- ○ Elements are stored in a sequential manner
- ○ Top of the stack contains the last inserted element

✓ 5) What is the outcome of the prefix expression +, -, *, 3, 2, /, 8, 4, 1? 1/1

- 12
- 11
- (●) 5 ✓
- 4

✓ 6) How many Queues are required to implement a Stack? 1/1

- 3
- (●) 2 ✓
- 1
- 4

✓ 7) Which one of the following is not the application of the Queue data structure? 1/1

- Resource shared between various systems
- Data is transferred asynchronously
- Load balancing
- (●) Balancing of symbols ✓

✗ 8) Which of the following option is true if implementation of Queue is from the linked list? 0/1

- ○ In enqueue operation, new nodes are inserted from the beginning and in dequeue operation, nodes are removed from the end.
- ◉ In enqueue operation, new nodes are inserted from the end and in dequeue operation, nodes are deleted from the beginning. ✗
- ○ In enqueue operation, new nodes are inserted from the end and in dequeue operation, nodes are deleted from the end.
- ○ Both a and b

Correct answer

- ◉ Both a and b

✓ 9) Which of the following statement is not true regarding the priority queue? 1/1

- ○ Processes with different priority can be easily handled
- ○ Easy to implement
- ◉ Deletion is easier ✓
- ○ None of the above

✓ 10) In the Deque implementation using singly linked list, what would be the time complexity of deleting an element from the rear end? 1/1

- ○ O(1)
- ○ O(n2)
- ◉ O(n) ✓
- ○ O(nlogn)

✓ 11) What would be the output after performing the following operations in a Deque? 1/1

```
Insertfront(10);
Insertfront(20);
Insertrear(30);
Insertrear(40);
Deletefront();
Insertfront(50);
Deleterear();
Display();
```

- ○ 10, 20, 30
- ◉ 50, 10, 30 ✓
- ○ 40, 20, 30
- ○ None of the above

✓ 12) The process of accessing data stored in a serial access memory is similar to manipulating data on a _____ 1/1

- ○ Heap
- ○ Binary Tree
- ○ Array
- ◉ Stack ✓

✓ 13) If circular queue is implemented using array having size MAX_SIZE in which array index starts with 0, front points to the first element in the queue, and rear points to the last element in the queue. Which one of the following conditions used to specify that the circular queue is empty? 1/1

- (●) Front=rear= -1 ✓
- ( ) Front=rear=0
- ( ) Front=rear+1
- ( ) None of the above

✓ 14) Which of the following statement is not true regarding the priority queue? 1/1

- ( ) Processes with different priority can be easily handled
- ( ) Easy to implement
- (●) Deletion is easier ✓
- ( ) None of the above

✗ 15) Which of the following is true about linked list implementation of queue? 0/1

- ○ In push operation, if new nodes are inserted at the beginning of linked list, then in pop operation, nodes must be removed from end
- ○ In push operation, if new nodes are inserted at the beginning, then in pop operation, nodes must be removed from the beginning
- ○ In push operation, if new nodes are inserted at the end, then in pop operation, nodes must be removed from end
- ◉ In push operation, if new nodes are inserted at the end, then in pop operation, nodes must be removed from beginning ✗

Correct answer

- ◉ In push operation, if new nodes are inserted at the beginning of linked list, then in pop operation, nodes must be removed from end

✗ 16) What are the applications of dequeue? 0/1

- ◉ A-Steal job scheduling algorithm ✗
- ○ Can be used as both stack and queue
- ○ To find the maximum of all sub arrays of size k
- ○ To avoid collision in hash tables

Correct answer

- ◉ To avoid collision in hash tables

✓ 17) Which application of stack is used to ensure that the pair of parentheses is properly nested? 1/1

- (●) Balancing symbols ✓
- ( ) Reversing a stack
- ( ) Conversion of an infix to postfix expression
- ( ) Conversion of an infix to prefix expression

✓ 18) When the corresponding end bracket/braces/parentheses is not found, what happens? 1/1

- ( ) The stack is popped
- ( ) Ignore the parentheses
- (●) An error is reported ✓
- ( ) It is treated as an exception

✓ 19) Which of the following statement is invalid with respect to balancing symbols? 1/1

- ( ) [(A+B) + (C-D)]
- ( ) [{A+B}-{C-[D+E]}]
- (●) ((A+B) + (C+D) ✓
- ( ) {(A+B) + [C+D]}

✗ 20) Minimum number of queues to implement stack is _____ 0/1

- ○ 3
- ○ Option 2
- ○ 1
- ◉ 2 ✗

Correct answer

- ◉ 1

DS_Queue_Mock_Test_2

Total points 20/20

Date:
Marks : 20 Marks
* Required

Email *

rkg16697@gmail.com

Roll Number *

1168

Name *

Rishikesh Gupta

✓ 1. A linear list of elements in which deletion can be done from one end (front) and insertion can take place only at the other end (rear) is known as _____ 1/1

- ○ Stack
- ○ Tree
- ○ Linked list
- ◉ Queue ✓

✓ 2. If the elements “A”, “B”, “C” and “D” are placed in a queue and are deleted one at a time, in what order will they be removed? 1/1

- (●) ABCD ✓
- ( ) DCBA
- ( ) DCAB
- ( ) ABDC

✓ 3. A normal queue, if implemented using an array of size MAX_SIZE, gets full when? 1/1

- (●) Rear = MAX_SIZE – 1 ✓
- ( ) Front = (rear + 1)mod MAX_SIZE
- ( ) Front = rear + 1
- ( ) Rear = front

✓ 4. Which one of the following is an application of Queue Data Structure? 1/1

- ( ) When a resource is shared among multiple consumers.
- ( ) When data is transferred asynchronously (data not necessarily received at same rate as sent) between two processes
- ( ) Load Balancing
- (●) All of the above ✓

✓ 5. If the MAX_SIZE is the size of the array used in the implementation of circular queue. How is rear manipulated while inserting an element in the queue? 1/1

- rear=(rear%1)+MAX_SIZE
- rear=rear%(MAX_SIZE+1)
- rear=(rear+1)%MAX_SIZE ✓
- rear=rear+(1%MAX_SIZE)

✓ 6. If the MAX_SIZE is the size of the array used in the implementation of circular queue, array index start with 0, front point to the first element in the queue, and rear point to the last element in the queue. Which of the following condition specify that circular queue is FULL? 1/1

- Front=rear= -1
- Front=(rear+1)%MAX_SIZE ✓
- Rear=front+1
- Rear=(front+1)%MAX_SIZE

✓ 7. A circular queue is implemented using an array of size 10. The array index starts with 0, front is 6, and rear is 9. The insertion of next element takes place at the array index. 1/1

- 0 ✓
- 7
- 9
- 10

✓ 8. If the MAX_SIZE is the size of the array used in the implementation of circular queue, array index start with 0, front point to the first element in the queue, and rear point to the last element in the queue. Which of the following condition specify that circular queue is EMPTY? 1/1

- ○ Front=rear=0
- ◉ Front = rear = -1 ✓
- ○ Front=rear+1
- ○ Front=(rear+1)%MAX_SIZE

✓ 9. In linked list implementation of a queue, front and rear pointers are tracked. Which of these pointers will change during an insertion into a NONEMPTY queue? 1/1

- ○ Only front pointer
- ◉ Only rear pointer ✓
- ○ Both front and rear pointer
- ○ None of the front and rear pointer

✓ 10. In linked list implementation of a queue, front and rear pointers are tracked. Which of these pointers will change during an insertion into EMPTY queue? 1/1

- ○ Only front pointer
- ○ Only rear pointer
- ◉ Both front and rear pointer ✓
- ○ None

✓ 11. In Queues, we can insert an element at ___ end and can delete an element at ____ end. 1/1

- (●) REAR, FRONT ✓
- ( ) FRONT,REAR
- ( ) TOP, BOTTOM
- ( ) BOTTOM, TOP

✓ 12. In Queue, ENQUEUE means____ whereas DEQUEUE refers____. 1/1

- (●) an insertion operation, a deletion operation. ✓
- ( ) End of the queue, defining a queue.
- ( ) Both A and B.
- ( ) None of the above are true.

✓ 13. Difference between stack and queue is ____ 1/1

- ( ) queue requires dynamic memory, but stack do not.
- ( ) stack requires dynamic memory, nut queue do not.
- (●) queue uses two ends of the structure, stack uses only one. ✓
- ( ) stack uses two ends of the structure, queue uses only one.

✓ 14. Which of the following is true about linked list implementation of queue? 1/1

- ( ) In enqueue (insert) operation, if new nodes are inserted at the beginning of linked list, then in dequeue (delete) operation, nodes must be removed from end.
- ( ) In enqueue (insert) operation, if new nodes are inserted at the end, then in dequeue (delete) operation, nodes must be removed from the beginning.
- (•) Both of the above ✓
- ( ) None of the above

✓ 15. A queue follows _____ 1/1

- (•) FIFO (First In First Out) principle ✓
- ( ) LIFO (Last In First Out) principle
- ( ) Ordered array
- ( ) Linear tree

✓ 16. Queues serve major role in _____ 1/1

- ( ) Simulation of recursion
- ( ) Simulation of arbitrary linked list
- (•) Simulation of limited resource allocation ✓
- ( ) Simulation of heap sort

✓ 17. How many stacks are needed to implement a queue. Consider the situation where no other data structure like arrays, linked list is available to you. 1/1

- ○ 1
- ◉ 2 ✓
- ○ 3
- ○ 4

✓ 18. In linked list implementation of queue, if only front reference is maintained, which of the following operation take worst case linear time? 1/1

- ○ Insertion
- ○ Deletion
- ○ To empty a queue
- ◉ Both a) and c) ✓

✓ 19. A normal queue, if implemented using an array of size MAX_SIZE, gets full when _____ 1/1

- ◉ Rear = MAX_SIZE − 1 ✓
- ○ Front = (rear + 1)mod MAX_SIZE
- ○ Front = rear + 1
- ○ Rear = front

✓ 20. Process of deleting an elements from the front end of queue is known as? 1/1

- (●) Dequeue ✓
- ( ) Enqueue
- ( ) Push
- ( ) Pop

DS_Searching_Sorting_Mock_Test_4

Total points 19/20

Date:
Marks : 20 Marks
* Required

Email *

rkg16697@gmail.com

Email *

rkg16697@gmail.com

Roll Number *

1168

Name *

Rishikesh Gupta

✓ 1. What is the frequency count of a nested quadratic loop? 1/1

- ○ f(n) = 1
- ○ f(n) = n
- ○ f(n) = logn
- ◉ f(n) = n^2 ✓

✓ 2. Arrange the following frequency counts from slowest to fastest : 1/1

- (●) f(1), f(logn), f(n), f(nlogn), f(n^2), f(2^n) ✓
- ( ) f(1), f(n), f(logn), f(nlogn), f(n^2), f(2^n)
- ( ) f(1), f(logn), f(nlogn), f(n), f(n^2), f(2^n)
- ( ) f(1), f(n), f(nlogn), f(logn), f(n^2), f(2^n)

✓ 3. The two key measures to find efficiency of an algorithm are: 1/1

- (●) Time and space ✓
- ( ) Capacity and Complexity
- ( ) Data and space
- ( ) Processor and memory

✗ 4. What is the time, space complexity of following code: ···/1

```
int a = 0, b = 0;
for (i = 0; i < N; i++) {
   a = a + rand();
}
for (j = 0; j < M; j++) {
   b = b + rand();
}
```

- ○ O(N * M) time, O(1) space
- ○ O(N + M) time, O(N + M) space
- ◉ O(N + M) time, O(1) space ✗
- ○ O(N * M) time, O(N + M) space

No correct answers

✓ 5. The complexity of Binary search algorithm is _____ 1/1

- ○ O(n)
- ◉ O(logn) ✓
- ○ O(n^2)
- ○ O(n log n)

✓ 6. Where is linear searching used? 1/1

- ○ When the list has only a few elements
- ○ When performing a single search in an unordered list
- ○ Used all the time
- ◉ When the list has only a few elements and When performing a single search in an unordered list ✓

✓ 7. Which of the following is not the required condition for a binary search algorithm? 1/1

- ○ The list must be sorted
- ○ There should be direct access to the middle element in any sublist
- ◉ There must be a mechanism to delete and/or insert elements in the list. ✓
- ○ Number values should only be present

✓ 8. _____ is the process of arranging the elements of a particular data structure in some logical order. 1/1

- ○ Merging
- ○ Insertion
- ○ Traversing
- ◉ Sorting ✓

✓ 9. You have to sort a list L consisting of a sorted list followed by a few 'random' elements. Which of the following sorting methods would be especially suitable for such a task? 1/1

- ○ Bubble sort
- ○ Selection sort
- ○ Quick Sort
- ◉ Insertion Sort ✓

✓ 10. The complexity of Bubble sort algorithm is ____ 1/1

- ○ O(n)
- ○ O(log n)
- ◉ O(n^2) ✓
- ○ O(n log n)

✓ 11. The worst case complexity of quick sort is _____ 1/1

- ○ O(n)
- ○ O(log n)
- ◉ O(n^2) ✓
- ○ O(n log n)

✓ 12. Which algorithm is having highest space complexity? 1/1

- ○ Bubble sort
- ○ Insertion Sort
- ○ Quick Sort
- ◉ Merge Sort ✓

✓ 13. The complexity of the sorting algorithm measures the _____ as a function of the number n of items to be sorter. 1/1

- ○ average time
- ◉ running time ✓
- ○ average-case complexity
- ○ case-complexity

✓ 14. _____ is putting an element in the appropriate place in a sorted list yields a larger sorted order list. 1/1

- ◉ Insertion ✓
- ○ Extraction
- ○ Selection
- ○ Distribution

✓ 15. _____ is rearranging pairs of elements which are out of order, until no such pairs remain. 1/1

- ○ Insertion
- ◉ Exchange ✓
- ○ Selection
- ○ Distribution

✓ 16. Which of the following sorting algorithms is closely related to shell sort? 1/1

- ○ Selection sort
- ○ Merge sort
- ◉ Insertion sort ✓
- ○ Bucket sort

✓ 17. Why is Shell sort called as a generalization of Insertion sort? 1/1

- ◉ Shell sort allows an exchange of far items whereas insertion sort moves elements by one position ✓
- ○ Improved lower bound analysis
- ○ Insertion is more efficient than any other algorithms
- ○ Shell sort performs internal sorting

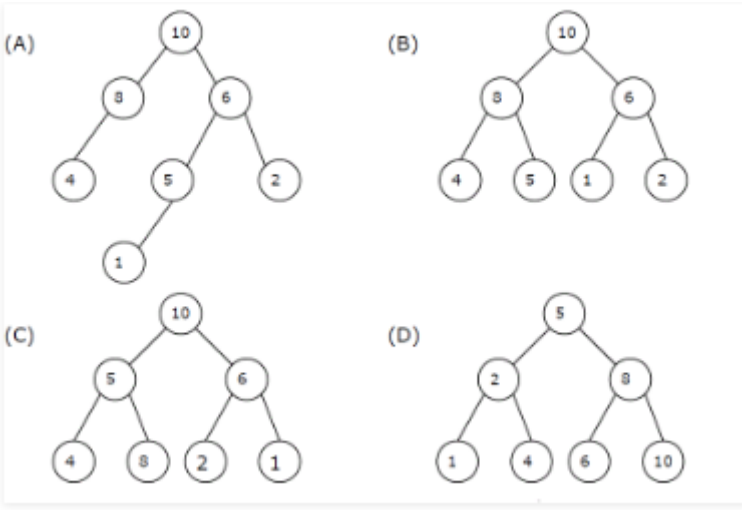
✓ 18. The time complexity of heap sort in worst case is ____ 1/1

- ○ O(logn)
- ○ O(n)
- ◉ O(nlogn) ✓
- ○ O(n^2)

✓ 19. Which of the following algorithm pays the least attention to the ordering of the elements in the input list? 1/1

- ○ Insertion sort
- ◉ Selection sort ✓
- ○ Quick sort
- ○ None

✓ 20. Which of the following is max-heap? 1/1



- [ ] A
- [x] B ✓
- [ ] C
- [ ] D

DS_Stack_Mock_Test_1

Total points 20/20

Date:
Marks : 20 Marks
* Required

Email *

rkg16697@gmail.com

Roll Number *

1168

Name *

Rishikesh

✓ 1. In a stack, if a user tries to remove an element from an empty stack it is called _____ 1/1

- (●) Underflow ✓
- ( ) Empty collection
- ( ) Overflow
- ( ) Garbage Collection

✓ 2. Entries in a stack are “ordered”. What is the meaning of this statement? 1/1

- ○ A collection of stacks is sortable
- ○ Stack entries may be compared with the ‘<‘ operation
- ○ The entries are stored in a linked list
- ◉ There is a Sequential entry that is one by one ✓

✓ 3. Which of the following is not the application of stack? 1/1

- ○ A parentheses balancing program
- ○ Tracking of local variables at run time
- ○ Compiler Syntax Analyzer
- ◉ Data Transfer between two asynchronous process ✓

✓ 4. What is the result of the following operation : Top (Push (S, X)) 1/1

- ◉ X ✓
- ○ Null
- ○ S
- ○ None

✓ 5. Which data structure is used for implementing recursion? 1/1

- ○ Queue
- ◉ Stack ✓
- ○ Array
- ○ List

✓ 6. Which of the following statement(s) about stack data structure is/are NOT correct? 1/1

- ○ Stack data structure can be implemented using linked list
- ○ New node can only be added at the top of the stack
- ◉ Stack is the FIFO data structure ✓
- ○ The last node at the bottom of the stack has a NULL link

✓ 7. Consider the linked list implementation of a stack. Which of the following node is considered as Top of the stack? 1/1

- ◉ First node ✓
- ○ Last node
- ○ Any node
- ○ Middle node

✓ 8. Consider the following operation performed on a stack of size 5. Push(1); Pop(); Push(2); Push(3); Pop(); Push(4); Pop(); Pop(); Push(5); After the completion of all operation, the no of element present on stack are _____ 1/1

- (●) 1 ✓
- ( ) 2
- ( ) 3
- ( ) 4

✓ 9. Which of the following is not an inherent application of stack? 1/1

- ( ) Reversing a string
- ( ) Evaluation of postfix expression
- ( ) Implementation of recursion
- (●) Job scheduling ✓

✓ 10. If the elements “A”, “B”, “C” and “D” are placed in a stack and are deleted one at a time, in what order will they be removed? 1/1

- ( ) ABCD
- (●) DCBA ✓
- ( ) DCAB
- ( ) ABDC

✓ 11. User perform following operations on stack of size 5 then -Push(1); pop(); push(2); push(3); pop(); push(2); pop(); pop(); push(4); pop(); pop(); push(5); Which of the following is correct statement for stack ? 1/1

- ○ Stack Operations will be performed Smoothly
- ◉ Underflow Occurs ✓
- ○ Overflow Occurs
- ○ None of the above

✓ 12. In liked representation of stack ..... holds the elements of the stack. 1/1

- ◉ INFO fields ✓
- ○ TOP fields
- ○ LINK fields
- ○ NULL fields

✓ 13. In linked representation of stack the null pointer of the last node in the list signals 1/1

- ○ Beginning of the stack
- ◉ Bottom of the stack ✓
- ○ Middle of the stack
- ○ In between some value

✓ 14. What happens when you push a new node onto a stack implemented using linked list? 1/1

- (●) The new node is placed at the front of the linked list ✓
- ( ) The new node is placed at the back of the linked list
- ( ) The new node is placed at the middle of the linked list
- ( ) No Changes happens

✓ 15. Which of the following name does not relate to stacks? 1/1

- (●) FIFO lists ✓
- ( ) LIFO lists
- ( ) Piles
- ( ) Push down lists

✓ 16. It is impossible to do ___ operation on empty stack. 1/1

- ( ) PUSH
- (●) POP ✓
- ( ) STATUS
- ( ) None

✓ 17. Consider the usual algorithm for determining whether a sequence of parentheses is balanced. The maximum number of parentheses that appear on the stack AT ANY ONE TIME when the algorithm analyzes: (() (())(()))? 1/1

- ○ 1
- ○ 2
- ◉ 3 ✓
- ○ 4 or more

✓ 18. Which of the following is true about linked list implementation of stack? 1/1

- ○ In push operation, if new nodes are inserted at the beginning of linked list, then in pop operation, nodes must be removed from end.
- ○ In push operation, if new nodes are inserted at the end, then in pop operation, nodes must be removed from the beginning.
- ○ Both of the above
- ◉ None of the above ✓

✓ 19. A single array A[1..MAXSIZE] is used to implement two stacks. The two stacks grow from opposite ends of the array. Variables top1 and top2 (topl< top 2) point to the location of the topmost element in each of the stacks. If the space is to be used efficiently, the condition for “stack full” is_____ 1/1

- (top1 = MAXSIZE/2) and (top2 = MAXSIZE/2+1)
- top1 + top2 = MAXSIZE
- (top1= MAXSIZE/2) or (top2 = MAXSIZE)
- top1= top2 -1 ✓

✓ 20 .Consider the following pseudocode that uses a stack? What is output for input "computer"? 1/1

```
declare a stack of characters
while ( there are more characters in the word to read )
{
  read a character
  push the character on the stack
}
while ( the stack is not empty )
{
  pop a character off the stack
  write the character to the screen
}
```

- computercomputer
- retupmoc ✓
- computer
- retupmocretupmoc

DS_Test3

Total points 20/20

Email *

rkg16697@gmail.com

Name *

Rishikesh Gupta

Roll No. *

1168

✓ 1) Which of the following options is not true about the Binary Search tree? 1/1

- ○ The value of the left child should be less than the root node
- ○ The value of the right child should be greater than the root node.
- ○ The left and right sub trees should also be a binary search tree
- ◉ None of the above ✓

✓ 2) What is/are the disadvantages of implementing tree using normal arrays? 1/1

- ○ difficulty in knowing children nodes of a node
- ○ difficult in finding the parent of a node
- ◉ have to know the maximum number of nodes possible before creation of trees ✓
- ○ difficult to implement

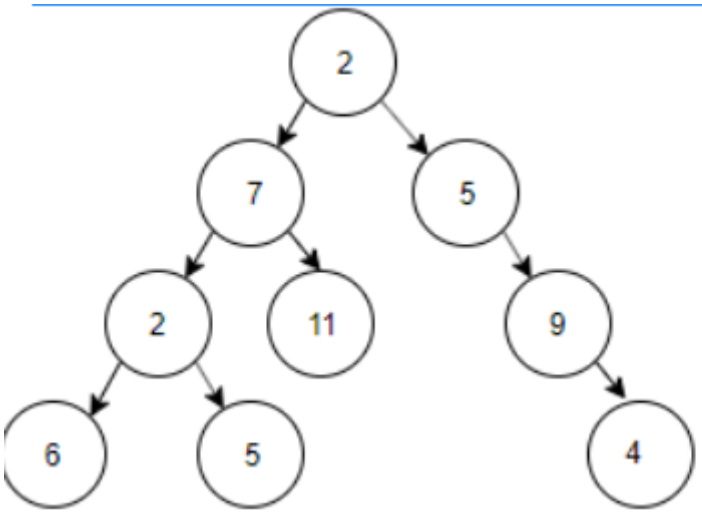
✓ 3) If the tree is not a complete binary tree then what changes can be made for easy access of children of a node in the array? 1/1

- ◉ every node stores data saying which of its children exist in the array ✓
- ○ no need of any changes continue with 2w and 2w+1, if node is at i
- ○ keep a seperate table telling children of a node
- ○ use another array parallel to the array with tree

✓ 4) Can a tree stored in an array using either one of inorder or post order or pre order traversals be again reformed? 1/1

- ○ Yes just traverse through the array and form the tree
- ◉ No we need one more traversal to form a tree ✓
- ○ No in case of sparse trees
- ○ Yes by using both inorder and array elements

✓ 5) For the tree below, write the pre-order traversal. 1/1



- (●) 2, 7, 2, 6, 5, 11, 5, 9, 4 ✓
- ( ) 2, 7, 5, 2, 6, 9, 5, 11, 4
- ( ) 2, 5, 11, 6, 7, 4, 9, 5, 2
- ( ) 2, 7, 5, 6, 11, 2, 5, 4, 9

✓ 6) To obtain a prefix expression, which of the tree traversals is used? 1/1

- ( ) Level-order traversal
- (●) Pre-order traversal ✓
- ( ) Post-order traversal
- ( ) In-order traversal

✓ 7) Consider the following data. The pre order traversal of a binary tree is A, B, E, C, D. The in order traversal of the same binary tree is B, E, A, D, C. The level order sequence for the binary tree is _____ 1/1

- ○ A, C, D, B, E
- ◉ A, B, C, D, E ✓
- ○ A, B, C, E, D
- ○ D, B, E, A, C

✓ 8) In postorder traversal of binary tree right subtree is traversed before visiting root. 1/1

- ◉ True ✓
- ○ False

✓ 9) What is the possible number of binary trees that can be created with 3 nodes, giving the sequence N, M, L when traversed in post-order. 1/1

- ○ 15
- ○ 3
- ◉ 5 ✓
- ○ 8

✓ 10) A binary search tree contains values 7, 8, 13, 26, 35, 40, 70, 75. Which one of the following is a valid post-order sequence of the tree provided the pre-order sequence as 35, 13, 7, 8, 26, 70, 40 and 75? 1/1

- ○ 7, 8, 26, 13, 75, 40, 70, 35
- ○ 26, 13, 7, 8, 70, 75, 40, 35
- ○ 7, 8, 13, 26, 35, 40, 70, 75
- ◉ 8, 7, 26, 13, 40, 75, 70, 35 ✓

✓ 11) What is a complete binary tree? 1/1

- ○ Each node has exactly zero or two children
- ○ A binary tree, which is completely filled, with the possible exception of the bottom level, which is filled from right to left
- ◉ A binary tree, which is completely filled, with the possible exception of the bottom level, which is filled from left to right ✓
- ○ A tree In which all nodes have degree 2

✓ 12) In a full binary tree if number of internal nodes is I, then number of leaves L are? 1/1

- ○ L = 2*I
- ◉ L = I + 1 ✓
- ○ L = I − 1
- ○ L = 2*I − 1

✓ 13) Which of the following is incorrect with respect to binary trees? 1/1

- Let T be a binary tree. For every k ≥ 0, there are no more than 2k nodes in level k
- Let T be a binary tree with λ levels. Then T has no more than 2λ − 1 nodes
- Let T be a binary tree with N nodes. Then the number of levels is at least ceil(log (N + 1))
- ◉ Let T be a binary tree with N nodes. Then the number of levels is at least floor(log (N + 1)) ✓

✓ 14) Which of the following is false about a binary search tree? 1/1

- The left child is always lesser than its parent
- The right child is always greater than its parent
- The left and right sub-trees should also be binary search trees
- ◉ In order sequence gives decreasing order of elements ✓

✓ 15) What are the worst case and average case complexities of a binary search tree? 1/1

- O(n), O(n)
- O(logn), O(logn)
- O(logn), O(n)
- ◉ O(n), O(logn) ✓

✓ 16) What are the conditions for an optimal binary search tree and what is its advantage? 1/1

- (●) The tree should not be modified and you should know how often the keys are accessed, it improves the lookup cost ✓
- ( ) You should know the frequency of access of the keys, improves the lookup time
- ( ) The tree can be modified and you should know the number of elements in the tree before hand, it improves the deletion time
- ( ) The tree should be just modified and improves the lookup time

✓ 17) Which of the following is the most widely used external memory data structure? 1/1

- ( ) AVL tree
- (●) B-tree ✓
- ( ) Red-black tree
- ( ) Both AVL tree and Red-black tree

✓ 18) B-tree and AVL tree have the same worst case time complexity for insertion and deletion. 1/1

- (●) True ✓
- ( ) False

✓ 19) Compression techniques can be used on the keys to reduce both space and time requirements in a B-tree. 1/1

- (●) True ✓
- ( ) False

✓ 20) Which of the following is true? 1/1

- (●) larger the order of B-tree, less frequently the split occurs ✓
- ( ) larger the order of B-tree, more frequently the split occurs
- ( ) smaller the order of B-tree, more frequently the split occurs
- ( ) smaller the order of B-tree, less frequently the split occurs

MOCK TEST 1 DS

Total points 20/20

Email *

rkg16697@gmail.com

Enter Roll No: *

1168

Enter Name: *

Rishikesh Gupta

✓ 1)Choose correct output for the following sequence of operations. push(5) push(8) pop() push(2) push(5) pop() pop() pop() push(1) pop() 1/1

- (●) 8 5 2 5 1 ✓
- ( ) 8 5 5 2 1
- ( ) 8 2 5 5 1
- ( ) 8 1 2 5 5

✓ 2)Stack can be implemented using _____ and _____ ? 1/1

- ○ Array and Binary Tree
- ○ Linked List and Graph
- ◉ Array and Linked List ✓
- ○ Queue and Linked List

✓ 3)Stack data structure cannot be used for 1/1

- ○ Implementation of Recursive Function
- ◉ Allocation Resources and Scheduling ✓
- ○ Reversing string
- ○ Evaluation of string in postfix form

✓ 4) Identify the data structure which allows deletions at both ends of the list but insertion at only one end 1/1

- ○ A Stack
- ○ B Priority queues
- ○ C Output restricted qequeue
- ◉ D Input restricted dequeue ✓

✓ 5)In a queue, the initial values of front pointer f rare pointer r should be ..... and ..... respectively 1/1

- A 0 and 1
- B 0 and -1 ✓
- C -1 and 0
- D 1 and 0 1

✓ 6) A data structure where elements can be added or removed at either end but not in the middle is called 1/1

- A stacks
- B queues C dequeue D linked lists
- C dequeue ✓
- D linked lists

✓ 7)Which of the following data structure can't store the nonhomogeneous data elements? 1/1

- A Arrays ✓
- B Stacks
- C Records
- D None of the above

✓ 8) A linear list in which the pointer points o 1/1

- (●) A singly linked list ✓
- ( ) B circular linked list
- ( ) C doubly linked list
- ( ) D none of the above

✓ 9)In a circular linked list 1/1

- (●) A.there is no beginning and no end. ✓
- ( ) B components are arranged hierarchically.
- ( ) C forward and backward traversal within the list is permitted.
- ( ) D components are all linked together in some sequential manner.

✓ 10)What data structure would you mostly likely see in a nonrecursive implementation of a recursive algorithm? 1/1

- ( ) A Trees
- ( ) B Linked list
- (●) C Stack ✓
- ( ) D Queue

✓ 11) Before inserting into stack one must check the condition..... 1/1

- A Overflow
- B Underflow ✓
- C Maximum elements
- D Existing elements

✓ 12)Deletion in the linked stack takes place by deleting..... 1/1

- A Beginning of the list
- B End of the list
- C Middle of the list
- D Node pointed by the start process. ✓

✓ 13)Let the following circular queue can accommodate maximum six elements with the following data front = 2 rear = 4 queue = _____; L, M, N, ___, ___ What will happen after ADD O operation takes place? 1/1

- a) front = 2 rear = 5 queue = _____; L, M, N, O, ___ ✓
- b) front = 3 rear = 5 queue = L, M, N, O, ___
- c) front = 3 rear = 4 queue = _____; L, M, N, O, ___
- d) front = 2 rear = 4 queue = L, M, N, O, ___

✓ 14)In linked list implementation of a queue, where does a new element be inserted? 1/1

- a) At the head of link list
- b) At the tail of the link list ✓
- c) At the centre position in the link list
- d) None

✓ 15) In the array implementation of circular queue, which of the following operation take worst case linear time? 1/1

- a) Insertion
- b) Deletion
- c) To empty a queue
- d) None ✓

✓ 16) If the MAX_SIZE is the size of the array used in the implementation of circular queue. How is rear manipulated while inserting an element in the queue? 1/1

- a) rear=(rear%1)+MAX_SIZE
- b) rear=rear%(MAX_SIZE+1)
- c) rear=(rear+1)%MAX_SIZE ✓
- d) rear=rear+(1%MAX_SIZE)

✓ 17)If the MAX_SIZE is the size of the array used in the implementation of circular queue, array index start with 0, front point to the first element in the queue, and rear point to the last element in the queue. Which of the following condition specify that circular queue is FULL? 1/1

- a) Front=rear= -1
- b) Front=(rear+1)%MAX_SIZE ✓
- c) Rear=front+1
- d) Rear=(front+1)%MAX_SIZE

✓ 18)In linked list implementation of a queue, front and rear pointers are tracked. Which of these pointers will change during an insertion into EMPTY queue? 1/1

- a) Only front pointer
- b) Only rear pointer
- c) Both front and rear pointer ✓
- d) None

✓ 19)Suppose a circular queue of capacity (n – 1) elements is implemented with an arrayof n elements. Assume that the insertion and deletion operation are carried outusing REAR and FRONT as array index variables, respectively. Initially, REAR = FRONT= 0. The conditions to detect queue full and queue empty are 1/1

- (a) Full: (REAR+1) mod n == FRONT, empty: REAR == FRONT ✓
- (b) Full: (REAR+1) mod n == FRONT, empty: (FRONT+1) mod n == REAR
- (c) Full: REAR == FRONT, empty: (REAR+1) mod n == FRONT
- (d) Full: (FRONT+1) mod n == REAR, empty: REAR == FRONT

✓ 20) Consider the usual algorithm for determining whether a sequence of parentheses isbalanced. What is the maximum number of parentheses that will appear on the stackAT ANY ONE TIME when the algorithm analyzes: (()(())(())) 1/1

- (a) 4
- (b) 3 ✓
- (c) 2
- (d) 6

MOCK TEST 3 DS USING JAVA

Total points 20/20

Email *

rkg16697@gmail.com

Enter Roll No: *

1168

Enter Name: *

Rishikesh Gupta

✓ 1) Merging k sorted tables into a single sorted table is called 1/1

- (●) A. k way merging ✓
- ( ) B. k th merge
- ( ) C. k+1 merge
- ( ) D. k-1 merge

✓ 2) If the number of record to be sorted large and the key is long, then ..... sorting can be efficient. 1/1

- A. Merge
- B. Heap
- C. Quick ✓
- D. Bubble

✓ 3)The function used to modify the way of sorting the keys of records is called 1/1

- A. Indexing function
- B. Hash function ✓
- C. Addressing function
- D. All of the above

✓ 4)The operation that combines the element is of A and B in a single sorted list C with n=r+s element is called 1/1

- A. Inserting
- B. Mixing ✓
- C. Merging
- D. Sharing

✓ 5)What is the worst case time for serial search finding a single item in an array 1/1

- ○ constant time
- ○ quadratic time
- ○ logarithmic time
- ◉ linear time ✓

✓ 6)What is the worst case time for binary search finding a single item in an array 1/1

- ○ constant time
- ◉ quadratic time ✓
- ○ logarithmic time
- ○ linear time

✓ 7)Which searching can be perform recursively 1/1

- ○ linear search
- ◉ both ✓
- ○ binary search
- ○ none

✓ 8)Suppose a complete binary tree has height h>0. The minimum no of leaf nodes possible in term of h is? 1/1

- A. 2h -1
- B. 2h -1 + 1
- C. 2h -1 ✓
- D. 2h +1

✓ 9)The post order traversal of binary tree is DEBFCA. Find out the pre order traversal. 1/1

- A. ABFCDE
- B. ADBFEC
- C. ABDECF ✓
- D. ABDCEF

✓ 10) In a binary tree, certain null entries are replaced by special pointers which point to nodes higher in the tree for efficiency. These special pointers are called 1/1

- A. Leaf
- B. Branch
- C. Path
- D. Thread ✓

✓ 11)If node N is a terminal node in a binary tree then its 1/1

- A. Right tree is empty
- B. Left tree is empty
- C. Both left & right sub trees are empty ✓
- D. Root node is empty

✓ 12)In linked representation of Binary trees LEFT[k] contains the ..... of at the node N, where k is the location. 1/1

- A. Data ✓
- B. Location and left child
- C. Right child address
- D. Null value

✓ 13)In-order traversing a tree resulted E A C K F H D B G; the pre-order traversal would return. 1/1

- A. FAEKCDBHG
- B. FAEKCDHGB ✓
- C. EAFKHDCBG
- D. FEAKDCHBG

✓ 14)The post order traversal of a binary tree is DEBFCA. Find out the pre order Traversal. 1/1

- A. ABFCDE
- B. ADBFEC
- C. ABDECF ✓
- D. ABDCEF

✓ 15)State true of false. i) A node is a parent if it has successor nodes.ii) A node is child node if out degree is one. 1/1

- True, True
- True, False ✓
- False, True
- False, False

✓ 16)State true or false. i) An empty tree is also a binary tree.ii) In strictly binary tree, the out-degree of every node is either o or 2. 1/1

- True, False
- False, True
- True, True ✓
- False, False

✓ 17)When inorder traversing a tree resulted E A C K F H D B G; the preorder traversal would return 1/1

- FAEKCDBHG
- FAEKCDHGB ✓
- EAFKHDCBG
- FEAKDCHBG

✓ 18)When representing any algebraic expression E which uses only binary operations in a 2-tree, 1/1

- the variable in E will appear as external nodes and operations in internal nodes ✓
- the operations in E will appear as external nodes and variables in internal nodes
- the variables and operations in E will appear only in internal nodes
- the variables and operations in E will appear only in external nodes

✓ 19)In a Heap tree 1/1

- Values in a node is greater than every value in left sub tree and smaller than right sub tree
- Values in a node is greater than every value in children of it ✓
- Both of above conditions applies
- None of above conditions applies

✓ 20)Let T be a binary search tree with 15 nodes. The minimum and maximum possible heights of T are:Note: The height of a tree with a single node is 0. 1/1

- ○ 4 and 15 respectively
- ◉ 3 and 14 respectively ✓
- ○ 4 and 14 respectively
- ○ 3 and 15 respectively