---------------------DATA STRUCTURE QUIZ-----------------------------

1. data structure can't store the non-homogeneous data elements?------ Ans) Arrays
2. data structure store the homogeneous data elements? ----- Ans) Records
3. non-linear data structure?------- Ans) Trees
4. Why ADT is called as abstract ?---------  ANS)Implementation details are hidden
5. In what kind of storage we can easily insert,delete,concatenate and rearange substrings?-----ANS)  Linked list
6. data structure which are indexed strucures?--------- Ans) Linked arrays
7. Non primitive data structures are those which define the set of ?-------- Ans) Derived elements
8. Which one of the following  is an application of the stack data structure?-- Ans) All of the above
9. Operations on data structure may be ?----- Ans) All of the above
10. Which of the following is not the component of data structure?---- Ans) None of the above
11. Q)  Which of the following is a true about Binary Trees

**(A)** Every binary tree is either complete or full.  
 **(B)** Every complete binary tree is also a full binary tree.  
 **(C)** Every full binary tree is also a complete binary tree.  
 **(D)** No binary tree is both complete and full.  
 **(E)** None of the above

Ans) None of the above

**Explanation:** A full binary tree (sometimes proper binary tree or 2-tree or strictly binary tree) is a tree in which every node other than the leaves has two children.

A complete binary tree is a binary tree in which every level, except possibly the last, is completely filled, and all nodes are as far left as possible.

1. -Q)  Following is C like pseudo code of a function that takes a number as an argument, and uses a stack S to do processing.

void fun(int n)

{

    Stack S;  // Say it creates an empty stack S

    while (n > 0)

    {

      // This line pushes the value of n%2 to stack S

      push(&S, n%2);

n = n/2;

    }   // Run while Stack S is not empty

    while (!isEmpty(&S))

      printf("%d ", pop(&S)); // pop an element from S and print it

}

Ans) Prints binary representation of n

1. Q)  What does the following function do for a given Linked List with first node as *head*?

void fun1(struct node\* head)

{

if(head == NULL)

return;

fun1(head->next);

printf("%d ", head->data);

}

Ans) Prints all nodes of linked list in reverse order

1. Q)  Following is C like pseudo code of a function that takes a Queue as an argument, and uses a stack S to do processing.

void fun(Queue \*Q)

{

    Stack S;  // Say it creates an empty stack S

   // Run while Q is not empty

    while (!isEmpty(Q))

    {

        // deQueue an item from Q and push the dequeued item to S

        push(&S, deQueue(Q));

   }

  // Run while Stack S is not empty

    while (!isEmpty(&S))

    {

      // Pop an item from S and enqueue the poppped item to Q

      enQueue(Q, pop(&S));

    }

}

Ans) Reverses the Q

1. Q)

Which of the following operations is not O(1) for an array of sorted data. You may assume that array elements are distinct.

Ans) Delete an element

1. Q)

What is common in three different types of traversals (Inorder, Preorder and Postorder)?

Ans) Left subtree is always visited before right subtree

1. Q)

Which one of the following is an application of Queue Data Structure?

Ans) All of the above

1. Q)

Which of the following points is/are true about Linked List data structure when it is compared with array

Ans) All of the above

1. Q)  What does the following function do for a given binary tree?

int fun(struct node \*root)

{

   if (root == NULL)

      return 0;

   if (root->left == NULL && root->right == NULL)

      return 0;

   return 1 + fun(root->left) + fun(root->right);

}

Ans) Counts internal nodes

1. Q)

Which one of the following is an application of Stack Data Structure?

Ans) All of the above

1. Q)

Following function is supposed to calculate the maximum depth or height of a Binary tree — the number of nodes along the longest path from the root node down to the farthest leaf node.

|  |  |  |
| --- | --- | --- |
|  | int maxDepth(struct node\* node)  {  if (node==NULL)  return 0;  else  {  /\* compute the depth of each subtree \*/  int lDepth = maxDepth(node->left);  int rDepth = maxDepth(node->right);  /\* use the larger one \*/  if (lDepth > rDepth)  return X;  else return Y;  }  } |  |
| What should be the values of X and Y so that the function works correctly? (A) X = lDepth, Y = rDepth  (B) X = lDepth + 1, Y = rDepth + 1  (C) X = lDepth – 1, Y = rDepth -1  (D) None of the above |  |
| ANS- B |  |

1. Q)

If arity of operators is fixed, then which of the following notations can be used to parse expressions without parentheses?

a) Infix Notation (Inorder traversal of a expression tree)  
b) Postfix Notation (Postorder traversal of a expression tree)  
c) Prefix Notation (Preorder traversal of a expression tree

(A) b and c

(B) Only b

(C) a, b and c

(D) None of them

ANS—(A)

23 Q) What is common in three different types of traversals (Inorder, Preorder and Postorder)?

(A) Root is visited before right subtree

(B) Left subtree is always visited before right subtree

(C) Root is visited after left subtree

(D) All of the above

(E) None of the above

Answer: (B)

24-Q)

The inorder and preorder traversal of a binary tree are d b e a f c g and a b d e c f

g, respectively. The postorder traversal of the binary tree is:

(A) d e b f g c a

(B) e d b g f c a

(C) e d b f g c a

(D) d e f g b c a

Answer: (A)

25 Q)

What are the main applications of tree data structure?

1) Manipulate hierarchical data  
2) Make information easy to search (see tree traversal).  
3) Manipulate sorted lists of data  
4) Router algorithms  
5) Form of a multi-stage decision-making, like Chess Game.  
6) As a workflow for compositing digital images for visual effects

**(A)** 1, 2, 3, 4 and 6  
**(B)** 1, 2, 3, 4 and 5  
**(C)** 1, 3, 4, 5 and 6  
**(D)** 1, 2, 3, 4, 5 and 6

**Answer:** **(D)**

**26 Q)**

Level of a node is distance from root to that node. For example, level of root is 1 and levels of left and right children of root is 2. The maximum number of nodes on level i of a binary tree is

In the following answers, the operator ‘^’ indicates power.  
**(A)** 2^(i-1)  
**(B)** 2^i  
**(C)** 2^(i+1)  
**(D)** 2^[(i+1)/2]

**Answer: (A)**

**27 Q)**

In a complete k-ary tree, every internal node has exactly k children or no child. The number of leaves in such a tree with n internal nodes is:

(A) nk

(B) (n – 1) k+ 1

(C) n( k – 1) + 1

(D) n(k – 1)

**Answer:** **(C)**

**28 Q)**

The maximum number of binary trees that can be formed with three unlabeled nodes is:

(A) 1

(B) 5

(C) 4

(D) 3

**Answer:** **(B)**

**29 Q)**

What is the time complexity of Build Heap operation. Build Heap is used to build a max(or min) binary heap from a given array. Build Heap is used in Heap Sort as a first step for sorting.

(A) O(nLogn)

(B) O(n^2)

(C) O(Logn)

(D) O(n)

**Answer:** **(D)**

**30 Q)**

What does the following function do for a given binary tree?

|  |
| --- |
| int fun(struct node \*root)  {  if (root == NULL)  return 0;  if (root->left == NULL && root->right == NULL)  return 0;  return 1 + fun(root->left) + fun(root->right);  } |

1. Counts leaf nodes
2. Counts internal nodes
3. Returns height where height is defined as number of edges on the path from root to deepest node
4. Return diameter where diameter is number of edges on the longest path between any two nodes.

**Answer:** **(B)**

31 Q)

Which of the following sorting algorithms can be used to sort a random linked list with minimum time complexity?

(A) Insertion Sort

(B) Quick Sort

(C) Heap Sort

(D) Merge Sort

**Answer:** **(D)**

32 Q)

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.

(A) 1

(B) 2

(C) 3

(D) 4

**Answer:** **(B)**

33 Q)

What is the worst case time complexity for search, insert and delete operations in a general Binary Search Tree?

(A) O(n) for all

(B) O(Logn) for all

(C) O(Logn) for search and insert, and O(n) for delete

(D) O(Logn) for search, and O(n) for insert and delete

**Answer: (A)**

34 Q)

In delete operation of BST, we need inorder successor (or predecessor) of a node when the node to be deleted has both left and right child as non-empty. Which of the following is true about inorder successor needed in delete operation?  
**(A)** Inorder Successor is always a leaf node  
**(B)** Inorder successor is always either a leaf node or a node with empty left child  
**(C)** Inorder successor may be an ancestor of the node  
**(D)** Inorder successor is always either a leaf node or a node with empty right child

**Answer:** **(B)**

35 Q)

Which of the following pairs of traversals is not sufficient to build a binary tree from the given traversals?  
**(A)** Preorder and Inorder  
**(B)** Preorder and Postorder  
**(C)** Inorder and Postorder  
**(D)** None of the Above  
  
  
**Answer:** **(B)**

36 Q)

What is the output of following function for start pointing to first node of following linked list?

1->2->3->4->5->6

void fun(struct node\* start)

{

if(start == NULL)

return;

printf("%d ", start->data);

if(start->next != NULL )

fun(start->next->next);

printf("%d ", start->data);

}

(A) 1 4 6 6 4 1

(B) 1 3 5 1 3 5

(C) 1 2 3 5

(D) 1 3 5 5 3 1

**Answer: (D)**

37 Q)

A priority queue can efficiently implemented using which of the following data structures? Assume that the number of insert and peek (operation to see the current highest priority item) and extraction (remove the highest priority item) operations are almost same.  
**(A)** Array  
**(B)** Linked List  
**(C)** Heap Data Structures like Binary Heap, Fibonacci Heap  
**(D)** None of the above  
  
  
**Answer:** **(C)**

38 Q)

Which of the following is true about linked list implementation of stack?  
**(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 end, then in pop operation, nodes must be removed from the beginning.  
**(C)** Both of the above  
**(D)** None of the above  
  
  
**Answer:** **(D)**

39 Q)

Which of the following is true about linked list implementation of queue?  
**(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 end, then in pop operation, nodes must be removed from the beginning.  
**(C)** Both of the above  
**(D)** None of the above  
  
  
**Answer:** **(C)**