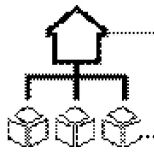


DATA STRUCTURE

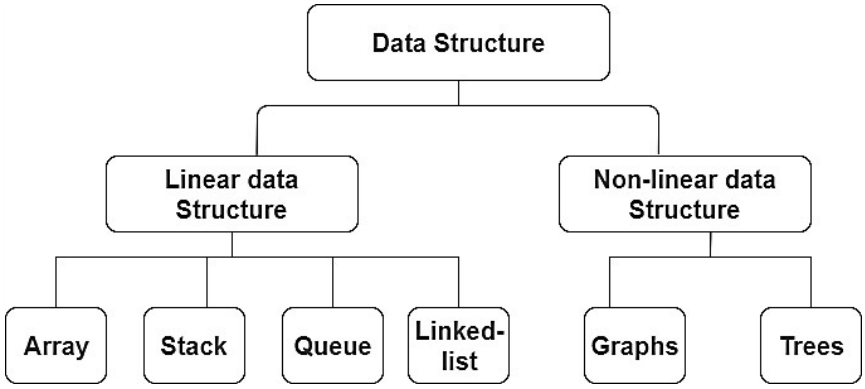
1. Which of the following data structure is non-linear type? *[Combined(O-IT/ICT)-2019]*

a) Strings b) Lists c) Stacks d) None of these **Ans: d**

Explanation: Data structures refers to the way data is organized and manipulated. It helps to find ways to make data access more efficient. When dealing with data structure, we not only focus on one piece of data, but rather different set of data and how they can relate to one another in an organized manner.

Linear Data Structures: The data structure where data items are organized sequentially or linearly where data elements attached one after another is called linear data structure. Data elements in a liner data structure are traversed one after the other and only one element can be directly reached while traversing. All the data items in linear data structure can be traversed in single run.

Non Linear Data Structures: The data structure where data items are not organized sequentially is called non linear data structure. In other words, A data elements of the non linear data structure could be connected to more than one elements to reflect a special relationship among them. All the data elements in non linear data structure can not be traversed in single run.



2. Which of the following is non-liner data structure?

a) Stacks b) List c) Strings d) Trees **Ans: d**

3. Which of the following data structure is non-linear type? *[Combined(O-IT/ICT)-2019]*

a)Strings b) Lists c) Stacks d) None of these **Ans: d**

4. Which is not linear? *[BB(AP)2016]*

a) linked list b) array c) graph d) None **Ans: c**

Explanation: **Commonly used data structures are,**

- ✓ list,
- ✓ arrays,
- ✓ stack,
- ✓ queues,
- ✓ graph,
- ✓ Tree.

Explanation: **The following operations are commonly performed on any data-structure,**

- ✓ **Insertion:** adding a data item.
- ✓ **Deletion:** removing a data item.
- ✓ **Traversal:** accessing and/or printing all data items.
- ✓ **Searching:** finding a particular data item.
- ✓ **Sorting:** arranging data items in a pre-defined sequence.

Explanation: **List out a few areas in which data structures are applied extensively?**

- ✓ Compiler Design,
- ✓ Operating System,
- ✓ Database Management System,
- ✓ Statistical analysis package,
- ✓ Numerical Analysis,
- ✓ Graphics,
- ✓ Artificial Intelligence,
- ✓ and Simulation.

5. What is the time complexity of following code:

```
int a = 0, i = N;
while (i> 0) {
   a += i;
   i /= 2;
}
```

a) O(N) b) O(Sqrt(N)) c)O(N / 2) d) O(log N) **Ans : d**

Explanation: We have to find the smallest x such that N / 2^x N x = log(N)

6. The complexity of Binary search algorithm is

a) O(n) b) O(log) c) O(n2) d) O(n log n) **Ans : b**

*Explanation:*The compexity of binary search is O(logn)

If the structure of a BST is a complete on balanced binary tree, it gives best performance and the time complexity of any operation (insert, delete and searching) is O (log n).

- ☞ Searching: Searching in a BST always starts at the root. We compare a data stored at the root with the key. If key is less than root then we search it to the left childs otherwise we search it to the right.
- ☞ Searching in a BST has O(h) worst-case runtime complexity, where h is the height of the

tree.

- Searching in a BST has O(log n) best-case runtime complexity, where n is the number of node of the tree.
- Insertion: BST has worst case complexity of O(n). In general, time complexity is O(h).
- Deletion: Deletion in binary tree has worst case complexity of O(n). In general, time complexity is O(h)

7. The complexity of merge sort algorithm is

a) O(n) b) O(log n) c) O(n2) d) O(n log n) **Ans :d**

Explanation:The worst case complexity for merge sort is O(nlogn)..

8. To measure Time complexity of an algorithm Big O notation is used which:

a) describes limiting behaviour of the function

b) characterises a function based on growth of function

c) upper bound on growth rate of the function

d) all of the mentioned **Ans :d**

Explanation:Big O notation describes limiting behaviour, and also gives upper bound on growth rate of a function.

- **Introduction**

  In computer science, the time complexity of an algorithm quantifies the amount of time taken by an algorithm to run as a function of the length of the string representing the input.

- **Big O notation**

  The time complexity of an algorithm is commonly expressed using big O notation, which excludes coefficients and lower order terms. When expressed this way, the time complexity is said to be described asymptotically, i.e as the input size goes to infinity.

  For example, if the time required by an algorithm on all inputs of size n is at most $5n^3 + 3n$, the asymptotic time complexity is $O(n^3)$. More on that later.

  Few more Examples:

  - ✓ $1 = O(n)$
  - ✓ $n = O(n^2)$
  - ✓ $\log(n) = O(n)$
  - ✓ $2 n + 1 = O(n)$

- **O(1) Constant Time:**

  An algorithm is said to run in constant time if it requires the same amount of time regardless of the input size.

  - ✓ Examples array: accessing any element
  - ✓ fixed-size stack: push and pop methods
  - ✓ fixed-size queue: enqueue and dequeue methods

- **O(n) Linear Time**

  An algorithm is said to run in linear time if its time execution is directly proportional to the input size, i.e. time grows linearly as input size increases.Consider the following examples,

below I am linearly searching for an element, this has a time complexity of O(n).

- **More Examples:**
  - ✓ Array: Linear Search, Traversing, Find minimum etc
  - ✓ ArrayList: contains method
  - ✓ Queue: contains method
- **O(log n) Logarithmic Time:**

  An algorithm is said to run in logarithmic time if it's time execution is proportional to the logarithm of the input size.

  Example: Binary Search

  Recall the **twenty question** game - the task is to guess the value of a hidden number in an interval. Each time you make a guess, you are told whether your guess is too high or too low. Twenty questions game implies a strategy that uses your guess number to halve the interval size. This is an example of the general problem-solving method known as binary search
- **O(n2) Quadratic Time**

  An algorithm is said to run in quadratic time if it's time execution is proportional to the square of the input size.

  Examples:
  - ✓ Bubble Sort
  - ✓ Selection Sort
  - ✓ Insertion Sort

9. If for an algorithm time complexity is given by O(1) then complexityof it is:

a) constant b) polynomial

c) exponential d) none of the mentioned **Ans :a**

Explanation:The growth rate of that function will be constant.

If for an algorithm time complexity is given by O(log2n) then complexity will:

a) constant b) polynomial

d) exponential c) none of the mentioned **Ans :d**

How many function call will be performed to execute the following recursive function?

```
void function(int N) {
          if(N==0)
          return;
          function(N+1)
       }
```

a) N
b) 2*N
c) Infinite
d) The answer can vary depending on the initial value of N

Ans.: d

Explanation: The growth rate of that function will be logarithmic therefore complexity will be logarithmic.

Linked List

A linked list is a linear data structure similar to arrays where each element is a separate object. Each element or node of a list comproses of two items? the data and a reference to the next node.

There are 3 types of linked list.

Singly Linked List: Every node stores address or reference of next node in list and the last node has next address or reference as NULL.

Doubly Linked List: There are two references associated with each node, one of the references points to the next node and other to the previous node. Advantage of this data structure is that we can traverse in both the directions and for deletion we don't need to have explicit access to previous node.

Circular Linked List: Circular linked list is a linked list where all nodes are connected to form a circle. There is no NULL at the end. A circular linked list can be a singly circular linked list or doubly circular linked list. Advantage of this data structure is that any node can be made as starting node. This is useful in implementation of circular queue in linked list.

11. A linear collection of data elements where the linear node is given by means of pointer is called?

a) Linked list b) Node list c) Primitive list d) Unordered list **Ans.: a**

Explanation: In Linked list each node has its own data and the address of next node. These nodes are linked by using pointers. Node list is an object that consists of a list of all nodes in a document with in a particular selected set of nodes.

12. In linked list each node contain minimum of two fields. One field is data field to store the data second field is?

a) Pointer to character b) Pointer to integer

c) Pointer to node d) Node **Ans.: c**

Explanation: Each node in a linked list contains data and a pointer (reference) to the next node. Second field contains pointer to node.

13. What would be the asymptotic time complexity to add a node at the end of singly linked list, if the pointer is initially pointing to the head of the list?

a) O(1) b) O(n) c) θ(n) d) θ(1) **Ans.: c**

Explanation: In case of a linked list having n elements, we need to travel through every node of the list to add the element at the end of the list. Thus asymptotic time complexity is θ(n).

14. Which of the following is false about a doubly linked list?

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 **Ans.: d**

Explanation: A doubly linked list has two pointers 'left' and 'right' which enable it to traverse in either direction. Compared to singly liked list which has only a 'next' pointer, doubly linked list requires extra space to store this extra pointer. Every insertion and deletion requires manipulation of two pointers, hence it takes a bit longer time. **Implementing doubly linked list involves setting both left and right pointers to correct nodes and takes more time than singly linked list.**

15. What is a memory efficient double linked list?
a) Each node has only one pointer to traverse the list back and forth
b) The list has breakpoints for faster traversal
c) An auxiliary singly linked list acts as a helper list to traverse through the doubly linked list
d) A doubly linked list that uses bitwise AND operator for storing addresses **Ans.: a**

Explanation: Memory efficient doubly linked list has only one pointer to traverse the list back and forth. The implementation is based on pointer difference. It uses bitwise XOR operator to store the front and rear pointer addresses. Instead of storing actual memory address, every node store the XOR address of previous and next nodes.

16. What is the worst case time complexity of inserting a node in a doubly linked list?
a) O(nlogn) b) O(logn) c) O(n) d) O(1) **Ans: c**

17. Which is correct for stack? *[BB(AP)2016]*
a) FIFO b) LIFO c) Both d) None **Ans: b**

18. Stack operations are *--[BB(AP)2016]*
a) delete, insertion b) insertion, delete c) push,pop d)pop, push **Ans: c**

19. What does 'stack underflow' refer to?
a) accessing item from an undefined stack
b) adding items to a full stack
c) removing items from an empty stack
d) index out of bounds exception **Ans.: c**

Explanation: Removing items from an empty stack is termed as stack underflow.
Stack is an abstract data type with a bounded(predefined) capacity. It is a simple data structure that allows adding and removing elements in a particular order. Every time an element is added, it goes on the top of the stack and the only element that can be removed is the element that is at the top of the stack, just like a pile of objects.

20. Stack operations are --*[BB(AP)2016]*
a) delete, insertion b) insertion, delete c) push,pop d)pop, push **Ans: c**

Explanation: **Basic features of Stack**

- ✓ Stack is an ordered list of similar data type.
- ✓ Stack is a LIFO(Last in First out) structure or we can say FILO(First in Last out).
- ✓ push() function is used to insert new elements into the Stack and pop() function is used to remove an element from the stack. Both insertion and removal are allowed at only one end of Stack called Top.
- ✓ Stack is said to be in Overflow state when it is completely full and is said to be in Underflow state if it is completely empty.

21. The data structure required to check whether an expression contains balanced parenthesis is? *[Combined(AP)-2018]*

a) Stack b) Queue c) Array d)Tree **Ans: a**

22. Applications of Stack

The simplest application of a stack is to reverse a word. You push a given word to stack - letter by letter - and then pop letters from the stack.

There are other uses also like:

- ✓ Parsing
- ✓ Expression Conversion(Infix to Postfix, Postfix to Prefix etc)

Application of Stack :

- ✓ Parsing
- ✓ Recursive Function
- ✓ Calling Function
- ✓ Expression Evaluation
- ✓ Expression Conversion
- ✓ Infix to Postfix
- ✓ Infix to Prefix
- ✓ Postfix to Infix
- ✓ Prefix to Infix
- ✓ Towers of hanoi

IMPLEMENTATION

A stack can be implemented in any one of the following two ways:

- ✓ Using an array.
- ✓ Using a linked list.

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 a ?

a) Queue b) Stack c) Tree d) Linked list **Ans.: a**

Explanation: Linear list of elements in which deletion is done at front side and insertion at rear side is called Queue. In stack we will delete the last entered element first.

A queue follows _____

a) FIFO (First In First Out) principle b) LIFO (Last In First Out) principle

c) Ordered array d) Linear tree **Ans.: a**

Explanation: Element first added in queue will be deleted first which is FIFO principle.

Which of the following is not the type of queue?

a) Ordinary queue b) Single ended queue
c) Circular queue d) Priority queue **Ans.: b**

Explanation: Queue always has two ends. So, single ended queue is not the type of queue.

Queues serve major role in _____

a) Simulation of recursion
b) Simulation of arbitrary linked list
c) Simulation of limited resource allocation
d) Simulation of heap sort **Ans.: c**

Explanation: Simulation of recursion uses stack data structure. Simulation of arbitrary linked lists uses linked lists. Simulation of resource allocation uses queue as first entered data needs to be given first priority during resource allocation. Simulation of heap sort uses heap data structure.

26. Circular Queue is also known as _____

a) Ring Buffer b) Square Buffer c) Rectangle Buffer d) Curve Buffer **Ans.: a**

Explanation: Circular Queue is also called as Ring Buffer. Circular Queue is a linear data structure in which last position is connected back to the first position to make a circle. It forms a ring structure.

27. A normal queue, if implemented using an array of size MAX_SIZE, gets full when

a) Rear = MAX_SIZE – 1 b) Front = (rear + 1)mod MAX_SIZE
c) Front = rear + 1 d) Rear = front **Ans.: a**

Explanation: When Rear = MAX_SIZE – 1, there will be no space left for the elements to be added in queue. Thus the queue becomes full.

There are many types of algorithm but the most fundamental types of algorithm are:

1. Recursive algorithms
2. Dynamic programming algorithm
3. Backtracking algorithm
4. Divide and conquer algorithm
5. Greedy algorithm
6. Brute Force algorithm
7. Randomized algorithm

Some Algorithms:

Dynamic Programming:

- ✓ Dynamic programming solves optimization problems by combining solutions to sub-problems.
- ✓ Dynamic programming is applicable when sub-problems are not independent
- ✓ Solve every sub-problem only once and store the answer in the table for use when it reappear.

- ✓ The key is to store the solutions of sub-problems to be reused in the future
- ✓ Example : *Matrix chain multiplication*

Divide-and-conquer approach:

- ✓ Partition the problem into independent sub-problems
- ✓ Solve the sub-problems recursively
- ✓ Combine solutions of sub-problems
- ✓ A divide-and-conquer approach will do more work than necessary
- ✓ Example: *Merge Sort*

Greedy algorithm:

- ✓ A greedy algorithm always makes the choice that looks best at the moment
- ✓ The hope is : a locally optimal choice will lead to a globally optimal solution
- ✓ For some problems, it works.
- ✓ Everyday examples:
- ✓ Playing cards
- ✓ Choosing a university

Given a sequence s={1,2,3,8,15,10}; which of the following algorithms will be fasted to sor this sequence in ascending order? ? *[Com (AP)- 2020]*

a) Bubble sort
b) Merge Sort
c) Quicksort
d) Heap Sort

Ans.: b

Which of the following techniques is popular for Data Compression? *[Com (AP)- 2020]*

a) Alpha-beta pruning
b) Checksum
c) Huffman Colding
d) Red black tree

Ans.: c

Explanation: Huffman coding is a lossless data compression algorithm. In this algorithm, a variable-length code is assigned to input different characters. The code length is related to how frequently characters are used. Most frequent characters have the smallest codes and longer codes for least frequent characters.There are mainly two parts. First one to create a Huffman tree, and another one to traverse the tree to find codes.

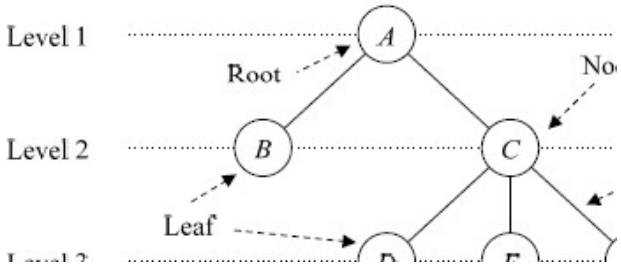
TREE

Tree is non-linear data structure designated at a special node calledroot and elements are arranged in levels without containing cycles.

Or

The tree is

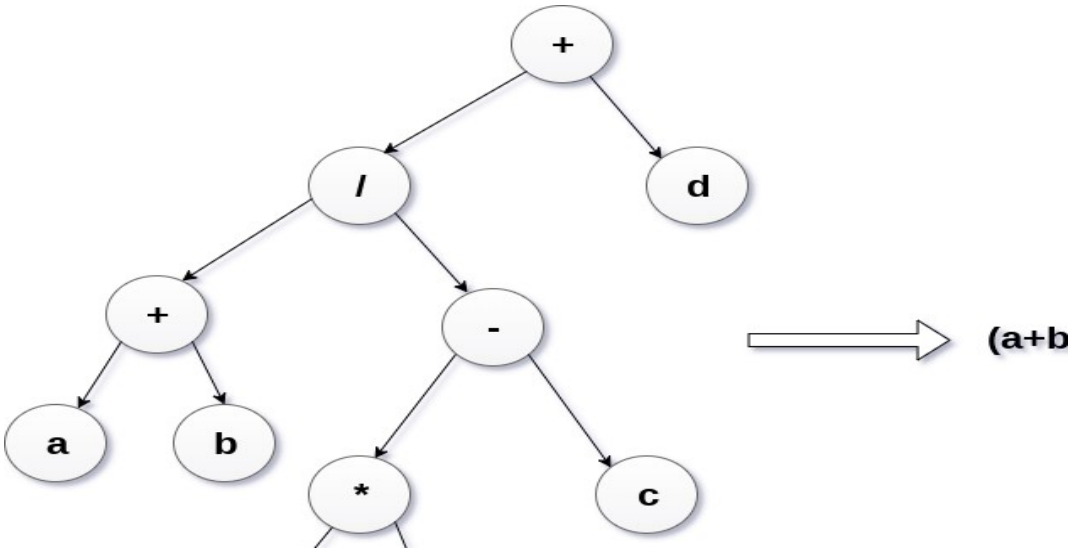
- ✓ Rooted at one vertex
- ✓ Contains no cycles
- ✓ There is a sequence of edges from any vertex to any other
- ✓ Any number of elements may connect to any node (includingroot)
- ✓ A unique path traverses from root to any node of tree
- ✓ The elements are arranged in layers



- ✓ **Node :** This is an individual element (i.e., data) that is shown as a circle.
- ✓ **Root :** This is the highest-level node. A tree structure has only one root.
- ✓ **Leaf :** This is the node that does not have any lower-level node.
- ✓ **Branch :** This is a line that connects to each node (including root, leaf).
- ✓ **Level :** This is the depth of hierarchy of a tree structure

Construct an expression tree by using the following algebraic expression.

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



BINARY TREE

It is a special type of tree where each node of tree contains either0 or 1 or 2 children.

Or

Binary Thee is either empty, or it consists of a root with two binarytrees called left-sub tree and right sub-tree of root (left or right orboth the sub trees may be empty)

Properties of binary tree

- ✓ Binary tree partitioned into three parts.
- ✓ First subset contains root of tree.
- ✓ Second subset is called left subtree.
- ✓ Another subset is called right subtree.
- ✓ Each subtree is a binary tree.
- ✓ Degree of any node is 0/1/2.
- ✓ The maximum number of nodes in a tree with height 'h' is $2^{h+1} - 1$
- ✓ The maximum number of nodes at level 'k' is $2^k - 1$.
- ✓ For any non-empty binary tree, the number of terminal nodes
- ✓ with n_2, nodes of degree 2 is $N_0 = n_2 + 1$
- ✓ The maximum number of nodes in a tree with depth d is $2^d - 1$.

Maximum how many nodes can be placed in a Binary Tree of N levels? *[Com(AP)-2020]*

a) 2^N
b) 2^{N-1}
c) $2^{N-1} - 1$
d) N^2

Ans.: c

Which algorithm will be most efficient to find out the shortest path between two given nodes in an undirected weighted graph? *[Com(AP)-2020]*

a) Breadth First Search
b) Depth First Search
c) Dijkstra's algorithm
d) Floyd-Warshall algorithm

Ans.: b

Some Graph Algorithm:

- ✓ **DFS:** The most useful graph algorithms are search algorithms. DFS (Depth First Search) is one of them
- ✓ **BFS:** BFS is another search algorithm (Breadth First Search). It is usually used to calculate the distances from a vertex *v* to all other vertices in unweighted graphs.
- ✓ **Shortest path:** Shortest path algorithms are algorithms to find some shortest paths in directed or undirected graphs.
- ✓ **Dijkstra:** This algorithm is a single source shortest path (from one source to any other vertices). Pay attention that you can't have edges with negative weight.
- ✓ **Floyd-Warshall:** Floyd-Warshal algorithm is an all-pairs shortest path algorithm using dynamic programming.
- ✓ **Bellman-Ford:** Bellman-Ford is an algorithm for single source shortest path where edges can be negative (but if there is a cycle with negative weight, then this problem will be NP).
- ✓ **SPFA:** SPFA (Shortest Path Faster Algorithm) is a fast and simple algorithm (single source) that its complexity is not calculated yet.

MST

- ✓ MST = Minimum Spanning Tree :) (if you don't know what it is, google it).
- ✓ **Best MST algorithms**:
- ✓ **Kruskal:** Kruskal's algorithm is a greedy algorithm, which helps us find the minimum spanning tree for a connected weighted graph, adding increasing cost arcs at each step.
- ✓ **Prim:** Prim's algorithm is a greedy algorithm, which helps us find the minimum spanning tree for a weighted undirected graph.

Previous Year Questions

29. Which data structure allow deleting data elements front and inserting at rear? *[Assistant programmer (ICB)- 2017]*

a) Stack b) Queue c) Dequeue d) Binary search tree **Ans:** b

30. The term push and pop related to *[Combined (AME)-2018]*

a) Array b) list c) stack d) all of this **Ans:** c

31. Which of the following name does not related to stacks? *[AP(ICB)- 2017, AP(HBFC, KB)-2018]*

a) FIFO List b) LIFO List c)Piles d) Push Down List **Ans:** a

Explanation: FIFO is related to Queue.

32. The term push and pop related to *[Combined (AME)-2018]*

a)Array b) list c) stack d) all of this **Ans:** c

33. Which of these data types is used by operating system to manage the Recursion in Java? *[Combined(IT/ICT-2018)]*

a)Array b)Stack c) Queue d) Tree **Ans:** b

34. Pushing an element into stack already having five elements and stack size of 5. Result in *[Combined (AP)-2018]*

a) Overflow b) Crash c) Underflow d) User flow **Ans:** a**7.**

35. The data structure required to check whether an expression contains balanced parenthesis is *[Combined (AP)-2018]*

Or, **The data structure required to evaluate a postfix expression is?** *[AP (ICB)- 2017]*

a) Stack b) Queue c) Array d)Tree

Ans: a

36. Find the correct arranged data after stack operation? *[BB (AP)-2016]*

push(1), push(2), pop, push(1), push(2), pop, pop, pop, push(2)

a) 2 2 1 1 2 b) 2 2 1 2 1 c) 2 2 2 2 1 d) 2 2 2 1 2

Operation	Stack	Pop Sequence
Push 1	1	
Push 2	1, 2	
Pop	1	2
Push 1	1, 1	
Push 2	1, 1, 2	
Pop	1, 1	2, 2
Pop	1	2, 2, 1
Pop	Empty	2, 2, 1, 1
Push 2	2	
Pop	Empty	2, 2, 1, 1, 2

Ans: a

9.

37. Stack operations are *[AP(SBL)-2016]*

a) delete, insertion b) insertion, delete c) push,pop d)pop, raer **Ans:** c

Some Practices Question

1. Process of inserting an element in stack is called _____

a) Create b) Push c) Evaluation d) Pop **Ans:** b

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

a) Managing function calls b) The stock span problem

c) Arithmetic expression evaluation d) All of the above **Ans:** d

3. What happens when you push a new node onto a stack?

a) The new node is placed at the front of the linked list

b) The new node is placed at the back of the linked list

c) The new node is placed at the middle of the linked list

d) No Changes happens **Ans:** a

4. In a stack, if a user tries to remove an element from empty stack it is called----

a) Underflow b) Empty collection

c) Overflow d) Garbage Collection **Ans:** a

5. Entries in a stack are "ordered". What is the meaning of this statement?

a) A collection of stacks is sortable

b) Stack entries may be compared with the '<' operation

c) The entries are stored in a linked list

d) There is a Sequential entry that is one by one **Ans:** d

6. The elements are removal from a stack in ……. order.

a) Reverse b) Hierarchical c) Alternative d) None **Ans:** a

7. Which of the following applications may use a stack?

a) A parentheses balancing program b) Tracking of local variables at run time

c) Compiler Syntax Analyzer d) All of the mentioned **Ans:** d

8. What is the value of the postfix expression 6 3 2 4 + – *

a) Something between -5 and -15 b) Something between 5 and -5

c) Something between 5 and 15 d) Something between 15 and 100

Explanation: On solving the postfix expression the answer comes out to 18 **Ans:** d

9. Here is an infix expression: 4 + 3*(6*3-12). Suppose that we are using the usual stack algorithm to convert the expression from infix to postfix notation. The maximum number of symbols that will appear on the stack AT ONE TIME during the conversion of this expression?

a) 1 b) 2 c) 3 d) 4 **Ans:** d

10. Which of the following is an application of stack?

a) finding factorial b) tower of Hanoi c) infix to postfix d) all of the above **Ans:** d

11. Stack is also called as

a) First in first out b) First in last out c) Last in last out d) Last in first out **Ans:** d

12. Which of the following data structure is linear type?

a) Stack b) Graph c) Trees d) Binary tree **Ans:** a

Data Structure All MCQ Questions

Multiple Choice Question and Answer:-

38. Which if the following is/are the levels of implementation of data structure

a) Abstract level b) Application level

c) Implementation level d) All of the above **Ans:** d

39. A binary search tree whose left subtree and right subtree differ in hight by at most 1 unit is called ……

a) AVL tree b) Red-black tree

c) Lemma tree d) None of the above **Ans:** d

40. --level is where the model becomes compatible executable code

a) Abstract level b) Application level

c) Implementation level d) All of the above **Ans:** c

41. Stack is also called as

a) Last in first out b) First in last out

c) Last in last out d) First in first out **Ans:** a

42. Which of the following is true about the characteristics of abstract data types?

i) It exports a type.

ii) It exports a set of operations

a) True, False b) False, True c) True, True d) False, False **Ans:** c

43. …… is not the component of data structure.

a) Operations b) Storage Structures

c) Algorithms d) None of above **Ans:** d

44. Which of the following is not the part of ADT description?

a) Data b) Operations

c) Both of the above d) None of the above **Ans:** d

45. Inserting an item into the stack when stack is not full is called ……. Operation and deletion of item form the stack, when stack is not empty is called ……..operation.

a) push, pop b) pop, push

c) insert, delete d) delete, insert **Ans:** a

46. ----Is a pile in which items are added at one end and removed from the other.

a) Stack b) Queue

c) List d) None of the above **Ans:** b

47. Which of the following data structure can't store the non-homogeneous data elements?

a) Arrays b) Records c) Pointers d) Stacks **Ans:** a

48. A ……. is a data structure that organizes data similar to a line in the supermarket, where the first one in line is the first one out.

a) Queue linked list b) Stacks linked list

c) Both of them d) Neither of them **Ans:** a

49. Which of the following is non-liner data structure?

a) Stacks b) List c) Strings d) Trees **Ans:** d

50. Herder node is used as sentinel in …..

a) Graphs b) Stacks c) Binary tree d) Queues **Ans:** c

51. Which data structure is used in breadth first search of a graph to hold nodes?

a) Stack b) queue c) Tree d) Array **Ans:** b

52. Identify the data structure which allows deletions at both ends of the list but insertion at only one end.

a) Input restricted dequeue b) Output restricted qequeue

c) Priority queues d) Stack **Ans:** a

53. Which of the following data structure is non linear type?
a) Strings b) Lists c) Stacks d) Graph **Ans:** d

54. Which of the following data structure is linear type?
a) Graph b) Trees c) Binary tree d) Stack **Ans:** d

55. To represent hierarchical relationship between elements, Which data structure is suitable?
a) Dequeue b) Priority c) Tree d) Graph **Ans:** c

56. A directed graph is ……. if there is a path from each vertex to every other vertex in the digraph.
a) Weakly connected b) Strongly Connected
c) Tightly Connected d) Linearly Connected **Ans:** b

57. In the …….. traversal we process all of a vertex's descendants before we move to an adjacent vertex.
a) Depth First b) Breadth First
c) With First d) Depth Limited **Ans:** a

58. State True of False.
i) Network is a graph that has weights or costs associated with it.
ii) An undirected graph which contains no cycles is called a forest.
iii) A graph is said to be complete if there is no edge between every pair of vertices.
a) True, False, True b) True, True, False
c) True, True, True d) False, True, True **Ans:** b

59. Match the following.
a) Completeness i) How long does it take to find a solution
b) Time Complexity ii) How much memory need to perform the search.
c) Space Complexity iii) Is the strategy guaranteed to find the solution when there in one.
a) a-iii, b-ii, c-i b) a-i, b-ii, c-iii c) a-iii, b-i, c-ii d) a-i, b-iii, c-ii **Ans:** c

60. The number of comparisons done by sequential search is ……
a) (N/2)+1 b) (N+1)/2 c) (N-1)/2 d) (N+2)/2 **Ans:** b

61. In ……, search start at the beginning of the list and check every element in the list.
a) Linear search b) Binary search
c) Hash Search d) Binary Tree search **Ans:** A

62. State True or False.
i) Binary search is used for searching in a sorted array.
ii) The time complexity of binary search is O(logn).
A) True, False B) False, True c) False, False d) True, True **Ans:** d

63. Which of the following is not the internal sort?
a) Insertion Sort b) Bubble Sort c) Merge Sort d) Heap Sort **Ans:** c

64. A graph is said to be …… if the vertices can be split into two sets V1 and V2 such there are no edges between two vertices of V1 or two vertices of V2.
a) Partite b) Bipartite c) Rooted d) Bisects **Ans:** b

65. In a queue, the initial values of front pointer f rare pointer r should be …….. and …….. respectively.
a) 0 and 1 b) 0 and -1 c) -1 and 0 d) 1 and 0 **Ans:** b

66. The advantage of …….. is that they solve the problem if sequential storage representation. But disadvantage in that is they are sequential lists.
a) Lists b) Linked Lists c) Trees d) Queues **Ans:** b

67. What will be the value of top, if there is a size of stack STACK_SIZE is 5
a) 5 b) 6 c) 4 d) None **Ans:** c

68. …… is not the operation that can be performed on queue.
a) Insertion b) Deletion c) Retrieval d) Traversal **Ans:** d

69. There is an extra element at the head of the list called a …….
a) Antinel b) Sentinel c) List header d) List head **Ans:** c

70. A graph is a collection of nodes, called ……. And line segments called arcs or …….. that connect pair of nodes.
a) vertices, edges b) edges, vertices
c) vertices, paths d) graph node, edges **Ans:** A

71. A …….. is a graph that has weights of costs associated with its edges.
a) Network b) Weighted graph c) Both A and B d) None A and B **Ans:** c

72. Which of the following is not the type of queue?
a) Ordinary queue b) Single ended queue
c) Circular queue d) Priority queue **Ans:** b

73. The property of binary tree is
a) The first subset is called left subtree
b) The second subtree is called right subtree
c) The root cannot contain NULL
d) The right subtree can be empty **Ans:** d

74. State true or false.
i) The degree of root node is always zero.
ii) Nodes that are not root and not leaf are called as internal nodes.
a) True, True b) True, False c) False, True d) False, False **Ans:** c

75. Which is/are the application(s) of stack
a) Function calls b) Large number Arithmetic
c) Evaluation of arithmetic expressions d) All of the above **Ans:** d

76. Which of the following data structures are indexed structures?
a. Linear arrays b. Linked lists c. Queue d. Stack **Ans:** a

77. Which of the following data structure store the homogeneous data elements?

a. Arrays b. Records c. Pointers d. Lists **Ans:** b

78. When new data are to be inserted into a data structure, but there is not available space; this situation is usually called ….

a. Underflow b. overflow c. houseful d. saturated **Ans:** b