**✅ MCQs (1 to 100) — Algorithm & Data Structure eDAC Sept 21**

**Q1.** Linked link are not superior to STL vectors  
a) True  b) False

**Q2.** Deleting a node in a linked list is a simple matter of using the delete operator to free the node’s memory  
a) True  b) False

**Q3.** The advantage of link list over array is  
a) Link list can grow and shrink in size during the time  
b) Less space is required for storing elements  
c) Both 1 and 2 are correct  
d) None of the above

**Q4.** Which one of the following algorithm is NOT an example of Divide and conquer technique  
a) Quick Sort  b) Merge Sort  c) Bubble Sort  d) Binary Search

**Q5.** The inorder traversal of some binary tree produces the sequence DBEAFC, and the postorder traversal of the same tree produced the sequence DEBFCA. Which of the following is correct preorder traversal sequence?  
a) DBAECF  b) ABEDFC  c) ABDECF  d) None of the above

**Q6.** How many cycles should be contained in a tree?  
a) 0  b) At least 1  c) Any number  d) None of the above

**Q7.** If graph G has no edges then corresponding adjacency matrix is  
a) Unit matrix  b) Zero matrix  c) Matrix with all 1’s  d) None of the above

**Q8.** What is not true for linear collision processing?  
a) It is easier to program  
b) It may include more collision  
c) It requires space for links  
d) All are true

**Q9.** Algorithms can be represented in various ways EXCEPT  
a) PROGRAMS  b) FLOWCHARTS  c) DECISION CHARTS  d) SPREADSHEET

**Q10.** The element at the root of heap is  
a) Largest  
b) Depending on type of heap it may be smallest or largest  
c) Smallest  
d) None of the above

**Q11.** The end at which a new element gets added to queue is called  
a) Front  b) Rear  c) Top  d) Bottom

**Q12.** Stack can be represented using  
a) Arrays  
b) Arrays or linked list  
c) Only linked list  
d) None of the above

**Q13.** A graph is said to be a tree, if it satisfies which of the properties:  
a) If it is connected and there are no cycles in the graph  
b) If it is not connected and there are cycles in the graph  
c) If it is connected and there are cycles in the graph  
d) None of the above

**Q14.** Hashing refers to the process of deriving  
a) A record key from storage address  
b) Storage address from a record key  
c) A floating-point code from a record key  
d) None of the above

**Q15.** (Same as Q5)  
Answer choices repeated

**Q16.** Which of the following is not an operation of queue?  
a) empty(Q)  
b) deque(Q, X)  
c) enque(Q, X)  
d) push(Q, X)

**Q17.** In an adjacency matrix parallel edges are given by  
a) Similar columns  b) Similar rows  c) Not representable  d) None of the above

**Q18.** A dynamic data structure where we can search for desired records in O(log₂n) time is  
a) Heap  b) Binary Search Tree  c) Circularly Linked List  d) Array

**Q19.** We can efficiently reverse a string using a  
a) Linear queue  b) Circular queue  c) Stack  d) Doubly linked list

**Q20.** A, B, C, D, E pushed to stack → 4 pop → inserted to queue → delete 2 → push back to stack → pop 1, result?  
a) A  b) B  c) C  d) D

**Q21.** The memory address of the first element of an array is called  
a) Floor address  b) Foundation address  c) First address  d) Base address

**Q22.** Memory address of fifth element of an array is calculated by:  
a) LOC(Array[5]) = Base(Array) + w(5 – lower bound)  
b) LOC(Array[5]) = Base(Array[5]) + (5 – lower bound)  
c) LOC(Array[5]) = Base(Array[4]) + (5 – upper bound)  
d) None of the above

**Q23.** Which of the following data structures are indexed structures?  
a) Linear arrays  b) Linked lists  c) Both of above  d) None of above

**Q24.** Which of the following is not the required condition for binary search algorithm?  
a) List must be sorted  
b) Direct access to the middle element  
c) Mechanism to delete and/or insert elements in list  
d) None of above

**Q25.** Which is not a limitation of binary search algorithm?  
a) Must use a sorted array  
b) Sorting is expensive with insertions and deletions  
c) Must access middle element directly  
d) Not efficient for >1000 data elements

**Q26.** Two dimensional arrays are also called  
a) Tables arrays  b) Matrix arrays  c) Both of above  d) None of above

**Q27.** A variable P is called pointer if  
a) P contains the address of an element in DATA  
b) P points to address of first element  
c) P can store only memory addresses  
d) P contains the DATA and the address of DATA

**Q28.** Which of the following cannot store non-homogeneous data elements?  
a) Arrays  b) Records  c) Pointers  d) None

**Q29.** Before deleting an element from list we ensure  
a) It is a list  
b) It is not invalid  
c) It is not an empty list  
d) It must be full

**Q30.** Indivisible items in a record are called  
a) Elementary items  b) Atoms  c) Scalars  d) All of above

**Q31.** Difference between linear array and record is  
a) Arrays: homogeneous data; Records: heterogeneous  
b) Records may not have natural ordering  
c) Records form hierarchical structure  
d) All of above

**Q32.** Which of the following is false?  
a) Arrays are dense and static  
b) Linked list nodes not stored adjacently  
c) Pointers store next data element  
d) Linked list nodes contain info + next pointer

**Q33.** Binary search algorithm cannot be applied to  
a) Sorted linked list  
b) Sorted binary trees  
c) Sorted linear array  
d) Pointer array

**Q34.** No space to insert new data is called  
a) Underflow  b) Overflow  c) Housefull  d) Saturated

**Q35.** In linked list START = NULL indicates  
a) Underflow  b) Overflow  c) Housefull  d) Saturated

**Q36.** Which of the following does not relate to stacks?  
a) FIFO lists  b) LIFO list  c) Piles  d) Push-down lists

**Q37.** Which of the following is two-way list?  
a) Grounded header list  
b) Circular header list  
c) Linked list with header and trailer nodes  
d) None of above

**Q38.** The term “push” and “pop” relates to  
a) Array  b) Lists  c) Stacks  d) All of above

**Q39.** A data structure where elements can be added/removed at ends but not in middle  
a) Linked lists  b) Stacks  c) Queues  d) Deque

**Q40.** Inorder traversal: E A C K F H D B G. Find preorder.  
a) FAEKCDBHG  
b) FAEKCDHGB  
c) EAFKHDCBG  
d) FEAKDCHBG

**Q41.** Which data structure allows deleting data elements from front and inserting at rear?  
a) Stacks  b) Queues  c) Deques  d) Binary search tree

**Q42.** Identify the data structure which allows deletions at both ends but insertion at only one end.  
a) Input-restricted deque  
b) Output-restricted deque  
c) Priority queues  
d) None of above

**Q43.** Which of the following data structure is non-linear type?  
a) Strings  b) Lists  c) Stacks  d) None of above

**Q44.** Which of the following data structure is linear type?  
a) Strings  b) Lists  c) Queues  d) All of above

**Q45.** To represent hierarchical relationship between elements, which data structure is suitable?  
a) Deque  b) Priority  c) Tree  d) All of above

**Q46.** A binary tree whose every node has either zero or two children is called  
a) Complete binary tree  
b) Binary search tree  
c) Extended binary tree  
d) None of above

**Q47.** The depth of a complete binary tree is given by  
a) Dn = n log₂n  
b) Dn = n log₂n+1  
c) Dn = log₂n  
d) Dn = log₂n+1

**Q48.** When representing any algebraic expression E in a 2-tree,  
a) Variables appear as external nodes, operations as internal nodes  
b) Operations appear as external, variables as internal  
c) All in internal nodes  
d) All in external nodes

**Q49.** A binary tree can easily be converted into a 2-tree  
a) Replace each empty subtree with a new internal node  
b) Insert internal nodes for non-empty node  
c) Insert external nodes for non-empty node  
d) Replace each empty subtree with a new external node

**Q50.** When converting binary tree to extended binary tree, all original nodes are  
a) Internal nodes on extended tree  
b) External nodes on extended tree  
c) Vanished on extended tree  
d) None of above

**Q51.** The post order traversal is DEBFCA. What is the preorder traversal?  
a) ABFCDE  b) ADBFEC  c) ABDECF  d) ABDCEF

**Q52.** Which sorting algorithm is divide-and-conquer type?  
a) Bubble sort  b) Insertion sort  c) Quick sort  d) All of above

**Q53.** An algorithm that calls itself is known as  
a) Sub algorithm  b) Recursion  c) Polish notation  d) Traversal algorithm

**Q54.** In a binary tree, null entries replaced by pointers to higher nodes are  
a) Leaf  b) Branch  c) Path  d) Thread

**Q55.** Inorder traversal yields sorted list in  
a) Binary trees  b) Binary search trees  c) Heaps  d) None of above

**Q56.** In a Heap tree  
a) Node > left, < right subtree  
b) Node > children  
c) Both a and b  
d) None

**Q57.** In a graph if e = [u, v], then u and v are  
a) Endpoints of e  
b) Adjacent nodes  
c) Neighbors  
d) All of above

**Q58.** A connected graph T without any cycles is  
a) A tree graph  b) Free tree  c) A tree  d) All of above

**Q59.** In a graph if e = (u, v) means  
a) u adjacent to v, but v not to u  
b) e begins at u, ends at v  
c) u is processor, v is successor  
d) Both b and c

**Q60.** If every node u is adjacent to every other node v, the graph is  
a) Isolated  b) Complete  c) Finite  d) Strongly connected

**Q61.** Two main measures for efficiency of algorithm are  
a) Processor and memory  
b) Complexity and capacity  
c) Time and space  
d) Data and space

**Q62.** Time factor is measured by  
a) Counting microseconds  
b) Counting key operations  
c) Counting number of statements  
d) Counting kilobytes of algorithm

**Q63.** Space factor is measured by  
a) Maximum memory needed  
b) Minimum memory needed  
c) Average memory needed  
d) Maximum disk space

**Q64.** Which case does not exist in complexity theory?  
a) Best case  b) Worst case  c) Average case  d) Null case

**Q65.** Worst case in linear search is when  
a) Item is in middle  
b) Item not in array  
c) Item is last  
d) Last or not in array

**Q66.** Average case in linear search is when  
a) Item in middle  
b) Item not in array  
c) Item is last  
d) Last or not in array

**Q67.** Complexity of average case is  
a) More complicated than worst  
b) Simpler than worst  
c) Sometimes simpler, sometimes not  
d) None of above

**Q68.** Complexity of linear search  
a) O(n)  b) O(log n)  c) O(n²)  d) O(n log n)

**Q69.** Complexity of binary search  
a) O(n)  b) O(log n)  c) O(n²)  d) O(n log n)

**Q70.** Complexity of bubble sort  
a) O(n)  b) O(log n)  c) O(n²)  d) O(n log n)

**Q71.** Complexity of merge sort  
a) O(n)  b) O(log n)  c) O(n²)  d) O(n log n)

**Q72.** Changing a variable’s value in one module by another module is  
a) Internal change  
b) Inter-module change  
c) Side effect  
d) Side-module update

**Q73.** Which is not a linear data structure?  
a) Arrays  b) Linked lists  c) Both  d) None

**Q74.** Which is linear data structure?  
a) Trees  b) Graphs  c) Arrays  d) None

**Q75.** Operation of processing each element in list is called  
a) Sorting  b) Merging  c) Inserting  d) Traversal

**Q76.** Finding location of element with given value is  
a) Traversal  b) Search  c) Sort  d) None

**Q77.** Arrays are best for  
a) Permanent data collection  
b) Changing data size  
c) Both above  
d) None

**Q78.** Linked lists are best for  
a) Permanent data collection  
b) Changing data size  
c) Both above  
d) None

**Q79.** *(Skipped in PDF)*

**Q80.** Each array declaration must give info about  
a) Name of array  
b) Data type  
c) First data  
d) Index set

**Q81.** Elements of array stored successively because  
a) Easy to calculate address from first  
b) Architecture forces it  
c) Both  
d) None

**Q82.** When is a linear queue empty?  
a) front > rear  
b) front == -1  
c) front > rear + 1  
d) rear == front + 1

**Q83.** Which is true about stacks and queues?  
i) Stack is full in sequential  
ii) Linear queue wastes memory  
iii) Circular queue full if rear = front + 1  
a) i & ii  b) i & iii  c) ii & iii  d) All

**Q84.** Queue-full condition for circular queue  
a) front == rear  
b) rear + 1 + front  
c) (rear + 1) % size == front  
d) None

**Q85.** In linked list, node consists of  
a) Data, link, header  
b) Only link  
c) Only data  
d) Data and link fields

**Q86.** In linked list  
a) Arrays used to hold list  
b) Each node links to next  
c) Links are array of pointers  
d) All above

**Q87.** *(Question missing)*

**Q88.** Which is NOT true for singly linked list?  
a) Nodes linked in one direction  
b) Last node points to NULL  
c) Search starts from first node  
d) Address of list is address of node

**Q89.** Main function header with command-line arguments:  
a) int main(int argc, char \*argv)  
b) int main(char \*argv, int argc)  
c) int main(int argc, char \*argv[])  
d) int main(char \*argv[], int argc)

**Q90.** Macro to display argument from variable arg function  
a) va\_arg  b) va\_list  c) va\_show  d) va\_start

**Q91.** Output of:

c

CopyEdit

float arr[]={12.5,5.4,7.3,21.6,8.7};

printf("%d\n", sizeof(arr)/sizeof(arr[0]));

a) 4  b) 5  c) 8  d) 20

**Q92.** Output of:

c

CopyEdit

for(j=1, sum=0; j<5; j++) sum += j; sum = j;

cout << sum;

a) 5  b) 10  c) Compilation error  d) 6

**Q93.** Best way to store frequency of scores > 50 (500 students, range 0–100)?  
a) Array of 50  
b) Array of 100  
c) Array of 500  
d) Dynamic array of 550

**Q94.** What is true about reference variable?  
a) Can’t be null  
b) Can’t be changed after assignment  
c) No need for dereference  
d) All of the above

**Q95.** Dynamic objects are stored in  
a) Code segment  b) Data segment  c) Heap  d) Stack

**Q96.** Output of code with const int\* Sample() returning address of const int a = 124  
a) Warning  b) Compilation error  c) Output 124  d) Garbage

**Q97.** Size of pointer in C++ on 32-bit architecture  
a) 1  b) 2  c) 4  d) Depends on datatype

**Q98.** Main features of OOP  
a) Encapsulation, Inheritance, Exception handling  
b) Inheritance, Polymorphism, Exception handling  
c) Encapsulation, Inheritance, Polymorphism  
d) Overloading, Inheritance, Polymorphism

**Q99.** Which cannot be declared virtual?  
a) Normal member function  
b) Constructor  
c) Destructor  
d) None

**Q100.** sizeof(Base) and sizeof(Derived) on 32-bit arch  
Base has: int i (private), int j (protected), int k (public)  
Derived has: int x, y, z  
a) 12, 12  
b) 12, 16  
c) 12, 24  
d) 4, 16

**Q101.** static\_cast can be applied at  
a) Compile time  b) Runtime  c) Linking time  d) None of these

**Q102.** Which inheritance type is used in class A: public B, public C {}?  
a) Multilevel  b) Multiple  c) Hybrid  d) Hierarchical

**Q103.** Which operator cannot be overloaded?  
a) []  b) ->  c) ?:  d) \*

**Q104.** Which STL container stores elements in adjacent memory?  
a) Vector  b) List  c) Set  d) Map

**Q105.** Incorrect statement about inline functions:  
a) Speeds up execution  
b) Slows down execution  
c) Increases code size  
d) Function can be inline without inline specifier

**Q106.** Which is not a member of a class?  
a) Static function  b) Friend function  c) Constructor  d) Virtual function

**Q107.** In which operator overloading is a dummy int passed?  
a) Post increment/decrement  
b) Pre increment/decrement  
c) Both  
d) None

**Q108.** Correct statement about abstract class:  
a) Cannot create object  
b) Pointer to abstract class can be created  
c) Reference to abstract class can be created  
d) All of the above

**Q109.** Not inherited during inheritance:  
a) Friend function  
b) Constructor  
c) Overloaded = operator  
d) All of the above

**Q110.** Output of the program:

cpp

CopyEdit

class myclass {

public: static int counter;

};

void main() {

cout << myclass::counter;

}

a) Output 0  b) Compilation error  c) Linking error  d) Garbage value

**Q111.** Purpose of template function:  
a) Allows same function with different types  
b) Hides name from linker  
c) Improves execution speed  
d) Enables better debugging

**Q112.** Which structure may overflow even if not full?  
a) Simple queue  b) Circular queue  c) Primary queue  d) Stack

**Q113.** Match sorting algorithms with complexities:  
a) Bubble sort  1) O(n log n)  
b) Insertion sort  2) O(n)  
c) Quick sort  3) O(n²)  
a) a=1 b=2 c=3  b) a=3 b=1 c=2  c) a=3 b=2 c=1  d) a=2 b=3 c=1

**Q114.** Null entries pointing to higher nodes in a binary tree are called  
a) Root  b) Node  c) Branch  d) Thread

**Q115.** BST with left & right subtree height difference ≤1 is called  
a) AVL tree  b) Red-black tree  c) Lemma tree  d) None

**Q116.** Which algorithm is not divide & conquer?  
a) Quick sort  b) Bubble sort  c) Merge sort  d) Binary search

**Q117.** Stack underflow occurs during  
a) Push  b) Pop  c) is\_full  d) None

**Q118.** Sorting algorithm with worst-case O(n log n)?  
a) Heap sort  b) Insertion sort  c) Selection sort  d) Bucket sort

**Q119.** Binary trees with 3 nodes with postorder A, B, C  
a) 3  b) 5  c) 7  d) 9

**Q120.** Binary tree with n leaf nodes (same level): non-leaf nodes?  
a) n - 1  b) log(n)  c) 2n  d) 2n - 1

**Q121.** Queue is used in  
a) Recursion  b) BFS  c) DFS  d) None

**Q122.** Design pattern in exception handling  
a) Chain of responsibility  b) Interpreter pattern  
c) Builder pattern  d) Adapter pattern

**Q123.** Design pattern to restrict class instantiation to one object  
a) Factory pattern  b) Builder pattern  
c) Prototype pattern  d) Singleton pattern

**Q124.** Object that exists between executions  
a) Global object  b) Persistent object  
c) Transient object  d) Delegate object

**Q125.** Design pattern to translate interface  
a) Proxy  b) Adapter  c) Façade  d) Bridge

**Q126.** Adapter, Bridge, Composite patterns are  
a) Creational  b) Structural  c) Behavioral  d) Interaction

**Q127.** Communication, sequence & timing diagrams are  
a) Behavior diagrams  b) Structure diagrams  
c) Activity diagrams  d) Interaction diagrams

**Q128.** Linked link are not superior to STL vectors  
a) True  b) False

**Q129.** Deleting a node using delete is simple  
a) True  b) False

**Q130.** Advantage of linked list over array  
a) Can grow/shrink  
b) Less space needed  
c) Both a and b  
d) None

**Q131.** Not divide-and-conquer algorithm  
a) Quick Sort  b) Merge Sort  c) Bubble Sort  d) Binary Search

**Q132.** Inorder: DBEAFC, Postorder: DEBFCA → Preorder?  
a) DBAECF  b) ABEDFC  c) ABDECF  d) None

**Q133.** Cycles in a tree?  
a) 0  b) At least 1  c) Any number  d) None

**Q134.** Graph with no edges → adjacency matrix is  
a) Unit matrix  b) Zero matrix  c) Matrix of 1s  d) None

**Q135.** What is not true for linear collision processing?  
a) Easier to program  
b) May include more collision  
c) Requires space for links  
d) All are true

**Q136.** Parallel edges in adjacency matrix  
a) Similar columns  b) Similar rows  c) Not representable  d) None

**Q137.** Element at root of heap  
a) Largest  
b) Smallest  
c) Depends on heap type  
d) None

**Q138.** End where queue adds new element  
a) Front  b) Rear  c) Top  d) Bottom

**Q139.** Preorder traversal of a tree  
a) ABDGCEHIF  b) ABDGHEICF  
c) ABDGFCIEH  d) None

**Q140.** A graph is a tree if  
a) Connected with no cycles  
b) Not connected and has cycles  
c) Connected and has cycles  
d) None

**Q141.** Hashing:  
a) Record key from address  
b) Storage address from record key  
c) Float code from record key  
d) None

**Q142.** (Same as Q132) Preorder?  
a) DBAECF  b) ABEDFC  c) ABDECF  d) None

**Q143.** What is not true for linear collision processing?  
(Same as Q135)

**Q144.** Parallel edges in adjacency matrix?  
(Same as Q136)

**Q145.** Output of:

cpp

CopyEdit

enum test { A=32, B, C };

cout << A << "," << B << "," << C;

a) 32,32,32  b) 32,33,34  c) 32,31,30  d) None

**Q146.** Structure with O(log₂n) search  
a) Heap  b) Binary Search Tree  c) Circular Linked List  d) Array

**Q147.** Efficient string reverse using  
a) Linear queue  b) Circular queue  c) Stack  d) Doubly linked list

**Q148.** Deleting node in linked list with delete is simple  
a) True  b) False

**Q149.** (Same as Q132 again) Preorder?  
a) DBAECF  b) ABEDFC  c) ABDECF  d) None

**Q150.** What is not true for linear collision processing?  
a) Easier to program  
b) May include more collision  
c) Requires space for links  
d) All are true

**Q151.** Parallel edges in adjacency matrix?  
a) Similar columns  b) Similar rows  c) Not representable  d) None

**Q152.** BST built with 6, 4, 2, 9, 5, 7, 0, 3, 1 → Inorder?  
a) 7 9 6 1 0 3 2 5 4  
b) 1 0 3 2 5 4 7 9 6  
c) 7 9 0 1 2 3 4 5 6  
d) 0 1 2 3 4 5 6 7 9

**Q153.** Measures of algorithm efficiency  
a) Data and space  
b) Processor and memory  
c) Complexity and capacity  
d) Time and space

**Q154.** Complexity of average case  
a) More complicated than worst  
b) Simpler than worst  
c) Sometimes simpler  
d) None

**Q155.** Time factor measured by  
a) Microseconds  b) Key operations  c) Number of statements  d) KBs

**Q156.** Space factor measured by  
a) Max memory  b) Min memory  c) Avg memory  d) Max disk space

**Q157.** Which case doesn't exist?  
a) Best  b) Worst  c) Average  d) Null

**Q158.** Running time of insertion sort  
a) O(n log n)  b) O(log n)  c) O(n)  d) O(n²)

**Q159.** Slowest sorting method  
a) Quick sort  b) Merge sort  c) Bubble sort  d) Heap sort

**Q160.** Efficiency order of bubble, selection, insertion  
a) bubble > selection > insertion  
b) insertion > selection > bubble  
c) merge = quick = heap  
d) None

**Q161.** Sorting by comparing & swapping with first element  
a) Quick sort  b) Selection sort  
c) Insertion sort  d) Merge sort

**Q162.** Card sorting by hand resembles  
a) Quick sort  b) Insertion sort  c) Selection sort  d) Merge sort

**Q163.** Best when list already sorted  
a) Merge sort  b) Quick sort  c) Insertion sort  d) Selection sort

**Q164.** Divide-and-conquer sorting algorithm  
a) Bubble sort  b) Insertion sort  c) Quick sort  d) All of above

**Q165.** Self-calling algorithm  
a) Sub-algorithm  b) Recursion  c) Polish notation  d) Traversal

**Q166.** Representation of data structure in memory  
a) Recursive  b) Abstract data type  
c) Storage structure  d) File structure

**Q167.** ADT is defined as model + set of  
a) Cardinality ops  b) Assignment ops  
c) Primitive ops  d) Structured ops

**Q168.** Algorithm with time complexities f(n), g(n) → total =  
a) f(n) \* g(n)  b) max(f(n), g(n))  
c) min(f(n), g(n))  d) f(n) + g(n)

**Q169.** Sorting books at end of day — best technique  
a) Bubble sort  b) Quick sort  c) Insertion sort  d) Selection sort

**Q170.** Merge sort recursive form  
a) T(n) = 2T(n/4) + n  
b) T(n) = 2T(n/2) + n  
c) T(n) = 2T(n/2) + 2  
d) T(n) = 2T(n/3) + n

**Q171.** You have a sorted array and now you are given an element to be placed in that array so that the resulting array is also sorted. The best sorting technique in this case is:  
a) Bubble sort  b) Selection sort  c) Insertion sort  d) Merge sort

**Q172.** Input to merge sort and quick sort is 6,5,4,3,2,1. Which is better?  
a) Merge sort  b) Quick sort  c) Cannot be decided

**Q173.** If memory available is less, which sort is better?  
a) Merge sort  b) Quick sort  c) Heap sort  d) All

**Q174.** Arrange heap, merge, quick sort in increasing space complexity:  
a) heap > merge > quick  
b) quick < heap < merge  
c) merge > quick > heap  
d) none

**Q175.** Quick sort is better because:  
a) O(n) runtime  b) θ(log n) space complexity

**Q176.** Quick sort running time largely depends on:  
a) Arrangement of elements  b) Pivot selection  
c) Small list  d) None

**Q177.** Heapify running time is:  
a) T(n) = T(2n/3) + Ω(1)  
b) T(n) = T(2n/2)  
c) T(n) = T(2n)  
d) None

**Q178.** Which statement about radix sort is true?  
a) LSD radix sort is stable  
b) MSD radix sort is stable  
c) None

**Q179.** LSD radix sort on 21, 86, 124, 33, 29, 163 — order before MSD?  
a) 21,29,86,33,124,163  
b) 21,124,29,33,163,86  
c) 21,29,124,163,33,86

**Q180.** Worst-case time and space complexity of radix sort is:  
a) O(nk), O(n+k)  b) O(n), O(n)  c) O(n log n), O(n log n)  d) None

**Q181.** Time complexity of counting sort:  
a) O(n + k)  b) O(n log n)  c) O(n²)  d) O(k²)

**Q182.** Counting sort is not comparison-based.  
a) True  b) False

**Q183.** Binary search can be applied to:  
a) Sorted list  b) Random list  c) Heap  d) Stack

**Q184.** Linked list is a:  
a) Linear data structure  
b) Non-linear data structure  
c) Matrix  
d) Array

**Q185.** In linked list, last node points to:  
a) Next node  b) Previous node  c) NULL  d) Root

**Q186.** Advantage of linked list over array:  
a) Dynamic size  
b) Easier insertion/deletion  
c) Efficient memory usage  
d) All of the above

**Q187.** Circular linked list uses last node to:  
a) Store NULL  
b) Point to first node  
c) Point to random node  
d) Point to header

**Q188.** In doubly linked list, each node contains:  
a) One pointer  
b) Two pointers  
c) Three pointers  
d) None

**Q189.** Which of the following is linear data structure?  
a) Array  b) Tree  c) Graph  d) Heap

**Q190.** Which of the following is non-linear data structure?  
a) Queue  b) Stack  c) Linked list  d) Tree

**Q191.** Maximum number of children for binary tree node?  
a) 1  b) 2  c) 3  d) 4

**Q192.** Which traversal outputs data in ascending order from BST?  
a) Preorder  b) Postorder  c) Inorder  d) None

**Q193.** Recursion uses which data structure?  
a) Queue  b) Heap  c) Stack  d) Linked list

**Q194.** In postorder traversal, root is visited:  
a) First  b) Middle  c) Last  d) None

**Q195.** In preorder traversal, order is:  
a) Root-Left-Right  b) Left-Root-Right  
c) Left-Right-Root  d) None

**Q196.** Complete binary tree of height h has how many nodes?  
a) 2^h - 1  b) 2^h  c) h^2  d) h

**Q197.** Graph is a collection of:  
a) Vertices and edges  
b) Nodes and branches  
c) Nodes and arrays  
d) Edges and matrices

**Q198.** BFS uses which data structure?  
a) Stack  b) Queue  c) Tree  d) Heap

**Q199.** DFS uses which data structure?  
a) Stack  b) Queue  c) Tree  d) Heap

**Q200.** Which traversal processes node before its children?  
a) Preorder  b) Postorder  c) Inorder  d) Level-order

**Q251.** What is the meaning of T in the class template?  
a) It must be an integer constant  
b) It is a string variable  
c) It is a placeholder for a type name  
d) It is a placeholder for a pointer value

**Q252.** In double order traversal:  
a) Every node is visited once  
b) Only root node is visited twice  
c) Some nodes are visited twice  
d) Every node is visited twice

**Q253.** What is the output of the following?

cpp

CopyEdit

char\* arr [] = {“C”,”C++”,”JAVA”,”VBA”};

char \*(\*ptr)[4] = &arr;

char<<++(\*ptr)[2];

a) Java b) C++ c) ava d) Compile time error

**Q254.** In recursion which data structure is used:  
a) Tree b) Linked List c) Array d) Stack

**Q255.** *(Not found or missing)*

**Q256.** Which of the following operators cannot be overloaded?  
a) = b) -> c) :: d) ==

**Q257.** Postfix equivalent of 4 $2\*3-3+8/4(1+1) is:  
a) 42$3*3-8/411+/+  
b) 42$3*3-84/11+/+  
c) 42$33\*-84/11+/+  
d) 42$3\*3-84/11++/

**Q258.** 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

**Q259.** Any node in the path from the root to the node is called  
a) Ancestor node b) Successor node  
c) Internal node d) None of the above

**Q260.** Which of the following is not a type of queue?  
a) Priority queue b) Circular queue  
c) Single ended queue d) Ordinary queue

**Q261.** A graph is a collection of nodes called \_\_\_ and arcs called \_\_\_:  
a) Vertices, paths b) Vertices, edges  
c) Graph node, edges d) Edges, vertices

**Q262.** In \_\_\_, search starts at the beginning of the list and checks each element.  
a) Binary search b) Hash search  
c) Linear search d) Tree search

**Q263.** In \_\_\_ traversal, we process all of a vertex’s descendants before moving to adjacent vertex.  
a) Depth Limited b) With First  
c) Breadth First d) Depth First

**Q264.** *(Not found or missing)*  
**Q265.** *(Not found or missing)*  
**Q266.** *(Not found or missing)*

**Q267.** Data structure to represent hierarchical relationship:  
a) Graph b) Tree c) Dequeue d) Priority

**Q268.** Which of the following is a linear data structure?  
a) Stack b) Graph c) Trees d) Binary tree

**Q269.** Header node is used as sentinel in:  
a) Queues b) Stacks c) Graphs d) Binary tree

**Q270.** Which data structure can't store non-homogeneous data elements?  
a) Arrays b) Stacks c) Records d) None

**Q271.** A binary search tree where left and right subtree differ by max 1 unit is:  
a) Lemma tree b) Red-black tree c) AVL tree d) None

**Q272.** A pile where items are added at one end and removed from other is:  
a) List b) Queue c) Stack d) Array

**Q273.** Which is non-linear data structure?  
a) Trees b) Stacks c) Strings d) All of the above

**Q274.** Comparisons in sequential search:  
a) (N/2)+1 b) (N+1)/2 c) (N-1)/2 d) (N-2)/2

**Q275.** Not a valid queue operation:  
a) Traversal b) Insertion c) Deletion d) Retrieval

**Q276.** Application(s) of stack:  
a) Function calls b) Arithmetic operations  
c) Evaluation of expressions d) All of the above

**Q277.** Which are indexed structures?  
a) Stack b) Linked lists c) Linear arrays d) None

**Q278.** Structure that stores homogeneous data:  
a) Lists b) Pointers c) Records d) Arrays

**Q279.** Linear arrays are also called:  
a) One-dimensional array b) Vertical array  
c) Horizontal array d) All of the above

**Q280.** Which does not keep track of the address of every element?  
a) Stack b) Queue c) String d) Linear array

**Q281.** Which of the following is not a linear data structure?  
a) Array b) Binary Tree c) Queue d) Stack

**Q282.** Which traversal gives postfix expression?  
a) Inorder b) Preorder c) Postorder d) Level-order

**Q283.** Traversal for prefix expression:  
a) Preorder b) Inorder c) Postorder d) None

**Q284.** A tree with no nodes is  
a) Empty tree b) Full tree c) Null tree d) Trivial tree

**Q285.** Binary tree with all levels filled except last is  
a) Balanced binary tree  
b) Complete binary tree  
c) Skewed binary tree  
d) Perfect binary tree

**Q286.** A tree with all internal nodes having 2 children and leaves at same level is  
a) Full binary tree  
b) Perfect binary tree  
c) Complete binary tree  
d) Degenerate binary tree

**Q287.** Binary tree with height h has max nodes:  
a) 2^h - 1 b) 2^(h+1) - 1 c) 2^h d) 2h - 1

**Q288.** Height of tree with 31 nodes in perfect binary tree is  
a) 4 b) 5 c) 6 d) 3

**Q289.** In binary search tree, right subtree has  
a) Equal values  
b) Smaller values  
c) Larger values  
d) Random values

**Q290.** Which traversal gives sorted output from BST?  
a) Preorder b) Inorder c) Postorder d) Level-order

**Q291.** Number of binary trees possible with 3 nodes:  
a) 5 b) 3 c) 4 d) 2

**Q292.** Time complexity of BST insert in worst case:  
a) O(log n) b) O(n) c) O(n log n) d) O(1)

**Q293.** Balanced tree avoids worst-case performance of  
a) O(n log n) b) O(log n) c) O(n²) d) O(n)

**Q294.** AVL trees are balanced using  
a) Insertion b) Deletion  
c) Rotations d) Traversals

**Q295.** Red-black tree is used to  
a) Ensure balance  
b) Ensure sorted order  
c) Ensure coloring  
d) None

**Q296.** Heap is what type of tree?  
a) Binary search tree  
b) Balanced tree  
c) Complete binary tree  
d) AVL tree

**Q297.** Heap is used for  
a) Searching b) Sorting  
c) Hashing d) Merging

**Q298.** Heap sort complexity:  
a) O(n) b) O(n²) c) O(n log n) d) O(log n)

**Q299.** Heap sort is:  
a) Internal sort b) External sort  
c) Both d) Not a sorting

**Q300.** Which is not a heap property?  
a) Shape property  
b) Order property  
c) Balance property  
d) Heap property

**Q301.** One can convert a binary tree into its mirror image by traversing it in  
a) Inorder b) Preorder c) Postorder d) None

**Q302.** Data structure required to evaluate postfix expression is  
a) Queue b) Stack c) Linked List d) All of the above

**Q303.** Sorting method best for nearly sorted list:  
a) Insertion sort b) Selection sort  
c) Quick sort d) Bubble sort

**Q304.** Accessing data in serial memory resembles  
a) Heap b) Queue c) Stack d) None

**Q305.** Postfix form of A*B+C/D is  
a) ABCD+/* b) AB\*CD/+ c) *AB/CD+ d) A*BC+/D

**Q306.** Linear collection with pointer reference:  
a) Linked list b) Node list  
c) Primitive list d) None

**Q307.** Data structure representation in memory is called  
a) Storage structure b) File structure  
c) ADT d) None

**Q308.** Goal of hashing is search with  
a) O(1) b) O(n²) c) O(log n) d) O(n log n)

**Q309.** Complexity of multiplying m×n and n×p matrices is  
a) np b) mn+p c) mn d) mnp

**Q310.** In undirected graph with n vertices and e edges, sum of degrees is  
a) 2n b) 2e c) (e² + 1)/2 d) (2n - 1)/2

**Q311.** Structure allowing delete from front and insert at rear is  
a) Stack b) Queue c) Deque d) BST

**Q312.** Breadth-first search uses  
a) Array b) Tree c) Stack d) Queue

**Q313.** Deletions at both ends, insertion at one end is  
a) Stack b) Priority queue  
c) Output-restricted deque d) Input-restricted deque

**Q314.** Which is non-linear data structure?  
a) Graph b) Stack c) List d) None

**Q315.** Initial value of front and rear in queue  
a) 0 and 1 b) 0 and -1 c) -1 and 0 d) 1 and 0

**Q316.** Extra element at list head used as  
a) Sentinel b) Antinel c) List head d) List header

**Q317.** Property of binary tree  
a) Root ≠ NULL  
b) Left = first subtree  
c) Right = second subtree  
d) Right subtree can be empty

**Q318.** Insertion not possible due to memory is called  
a) Overflow b) Underflow c) Housefull d) Memory full

**Q319.** Structure where insertion/deletion at ends only  
a) Stack b) Queue c) Deque d) Linked list

**Q320.** Logical adjacency via pointers:  
a) Stack b) Queue c) Pointer d) Linked allocation

**Q321.** Binary search cannot be applied on  
a) Pointer array b) Sorted array  
c) Sorted BST d) Sorted linked list

**Q322.** Sorting technique used by card sorters:  
a) Quick b) Heap c) Insertion d) Radix sort

**Q323.** Pointer-based memory allocation is  
a) Static b) Heap c) Stack d) Queue

**Q324.** Address part of a node stores  
a) Actual data b) Next node address  
c) Both d) None

**Q325.** Stack deletion follows  
a) FIFO b) FILO c) LIFO d) LILO

**Q326.** Operation checking for stack overflow  
a) Pop b) Push c) Traverse d) Search

**Q327.** End operation in stack  
a) Pop b) Traverse c) Delete d) Insert

**Q328.** Queue follows  
a) LIFO b) FIFO c) FILO d) LILO

**Q329.** Linear queue uses  
a) Two pointers b) One pointer  
c) Pointer and counter d) Three pointers

**Q330.** Linked list where last points to head  
a) Doubly list b) Circular list  
c) Reverse list d) General list

**Q331.** Sorted linked list is best for  
a) Stack b) Queue c) Priority queue d) Graph

**Q332.** AVL trees are  
a) Height-balanced BSTs b) Weight-balanced  
c) Skewed BSTs d) None

**Q333.** Postorder traversal of BST gives  
a) Prefix b) Suffix c) Infix d) All

**Q334.** DFS uses  
a) Queue b) Stack c) Linked list d) Tree

**Q335.** Preorder traversal order  
a) Root–Left–Right b) Left–Root–Right  
c) Left–Right–Root d) Root–Right–Left

**Q336.** Which structure is dynamic  
a) Array b) Stack c) Queue d) Linked list

**Q337.** Binary search requires  
a) Random b) Sorted c) Reversed d) Duplicates

**Q338.** Max children in binary tree  
a) 1 b) 2 c) 3 d) 4

**Q339.** Graph traversal for AI  
a) BFS b) DFS c) Both d) None

**Q340.** Tree with at most 2 children  
a) Binary tree b) General tree  
c) AVL tree d) Heap

**Q341.** Stack insertion is  
a) Push b) Pull c) Pop d) All

**Q342.** Stack deletion is  
a) Push b) Pull c) Pop d) All

**Q343.** Check before inserting to stack  
a) Overflow b) Underflow c) Max size d) Empty

**Q344.** Linked stack deletion occurs at  
a) Beginning b) End c) Middle d) Tail

**Q345.** REAR pointer is increased when  
a) Merge b) Insert c) Traverse d) Delete

**Q346.** Operation on every element in list  
a) Merge b) Traverse c) Insert d) Sort

**Q347.** Sequential binary tree uses  
a) Array with pointer b) Single array  
c) 2D array d) 3D array

**Q348.** Node with 0 children in 2-tree  
a) Outer node b) External node  
c) Outside node d) Leaf node

**Q349.** Node with 2 children in extended binary tree  
a) Inner node b) Internal node  
c) Domestic node d) Inside node

**Q350.** A line from one node to another is  
a) Route b) Arrow c) Edge d) Path

**Q351.** Which of the following sorting algorithms does not have a worst-case time complexity of O(n²)?  
a) Insertion sort b) Quick sort c) Bubble sort d) Merge sort

**Q352.** In a circular linked list  
a) No beginning and no end  
b) Hierarchical arrangement  
c) Forward and backward traversal permitted  
d) Components are sequentially linked

**Q353.** Quick sort uses which design technique?  
a) Overflow b) Backtracking c) Dynamic programming d) Divide and conquer

**Q354.** Which data structure is used to check balanced parentheses?  
a) Stack b) Queue c) Tree d) Array

**Q355.** In non-recursive version of recursive algorithm, which structure is used?  
a) Trees b) Linked list c) Stack d) Queue

**Q356.** Leaf nodes in a complete binary tree of depth d:  
a) 2ᵈ b) 2ᵈ⁻¹ + 1 c) 2ᵈ⁺¹ + 1 d) 2ᵈ⁺¹

**Q357.** If preorder and postorder of a binary tree are same, max nodes possible:  
a) 1 b) 2 c) 3 d) Any number

**Q358.** An almost complete binary tree has  
a) Leaves at level d or d–1  
b) If right descendant at d, then left also at d  
c) Both A & B  
d) None

**Q359.** Sequence of consecutive edges in binary tree is called:  
a) Path b) Rotate c) Two-way d) Connecting lines

**Q360.** Adjacency matrix cannot show:  
a) Nodes b) Edges c) Parallel edges d) Direction of edges

**Q361.** Which is not a queue operation?  
a) Traversal b) Retrieval c) Deletion d) Insertion

**Q362.** A linear list with last node pointing to first is called:  
a) Singly list b) Double list c) Circular list d) None

**Q363.** Binary tree with all nodes having 0 or 2 children:  
a) Complete b) Binary search tree c) Extended d) Strictly binary tree

**Q364.** Number of NULL links in binary tree with n nodes:  
a) n b) n+1 c) n-1 d) 2n

**Q365.** Inorder traversal:  
a) Left, Root, Right b) Root, Left, Right  
c) Right, Left, Root d) Left, Right, Root

**Q366.** A complete binary tree is  
a) All levels full except possibly last  
b) All leaves at same level  
c) Every node has 0 or 2 children  
d) None

**Q367.** Data structure ideal for recursion:  
a) Array b) Queue c) Stack d) Tree

**Q368.** Dijkstra’s algorithm finds:  
a) Shortest path b) Longest path  
c) Depth first path d) Minimum spanning tree

**Q369.** Data structure for implementing recursion:  
a) Linked list b) Queue c) Tree d) Stack

**Q370.** How many leaves in a full binary tree with n internal nodes?  
a) n b) n + 1 c) 2n d) log n

**Q371.** Which is not a graph traversal method?  
a) BFS b) DFS c) Backtracking d) Euler tour

**Q372.** How many edges in a tree with n vertices?  
a) n – 1 b) n c) n + 1 d) 2n

**Q373.** What is a node with no children called?  
a) Internal node b) External node  
c) Null node d) End node

**Q374.** Time complexity of searching in balanced BST  
a) O(1) b) O(log n) c) O(n) d) O(n²)

**Q375.** When to use hash table over binary search tree  
a) When many insertions are required  
b) When quick search is required  
c) When deletion is not needed  
d) For sorting

**Q376.** Which traversal gives sorted order in BST  
a) Preorder b) Postorder c) Inorder d) Level-order

**Q377.** A stack is a \_\_\_ list.  
a) LIFO b) FIFO c) FILO d) None

**Q378.** A queue is a \_\_\_ list.  
a) FIFO b) FILO c) LIFO d) None

**Q379.** Circular queue overcomes limitation of  
a) Linear queue  
b) Deque  
c) Stack  
d) Binary heap

**Q380.** Which is not an application of stack?  
a) Parenthesis checker  
b) Expression evaluation  
c) Function call tracking  
d) Scheduling processes

**Q381.** In linked list, insertion is efficient at  
a) Beginning b) Middle  
c) End d) Random

**Q382.** Stack underflow occurs when  
a) Pushing into full stack  
b) Popping from empty stack  
c) Searching in empty stack  
d) None

**Q383.** Hashing uses \_\_\_ to calculate address.  
a) Heap b) Stack c) Hash function d) Array

**Q384.** AVL tree balance factor values are  
a) -1, 0, 1 b) -2 to 2 c) 0, 1 d) -1, 1

**Q385.** Breadth-first traversal uses  
a) Stack b) Queue c) Tree d) Graph

**Q386.** Graph with no cycles  
a) Tree b) Directed graph  
c) DAG d) Connected graph

**Q387.** A stack is implemented using  
a) Queue b) Tree c) Array or linked list d) Graph

**Q388.** Minimum edges in connected graph with n nodes:  
a) n b) n–1 c) n+1 d) n²

**Q389.** Which tree traversal is non-recursive using stack?  
a) Postorder b) Inorder c) Preorder d) Level-order

**Q390.** Which queue allows insert/delete at both ends?  
a) Simple queue b) Priority queue  
c) Deque d) Circular queue

**Q391.** Adjacency list is preferred when  
a) Graph is dense  
b) Graph is sparse  
c) Space doesn’t matter  
d) Matrix operations needed

**Q392.** Which graph traversal uses queue?  
a) DFS b) BFS c) Preorder d) Postorder

**Q393.** Complete binary tree can be efficiently represented using  
a) Linked list b) 2D array  
c) Array d) Queue

**Q394.** Binary tree with max height is  
a) Balanced tree b) Skewed tree  
c) AVL tree d) Heap

**Q395.** In stack, the last inserted element is  
a) Deleted first b) Deleted last  
c) Never deleted d) Random

**Q396.** Sorting technique using heap  
a) Merge sort b) Quick sort  
c) Heap sort d) Radix sort

**Q397.** Which is an example of divide and conquer?  
a) Bubble sort b) Quick sort  
c) Selection sort d) Insertion sort

**Q398.** In doubly linked list, each node has  
a) One link b) Two links  
c) Three links d) Four links

**Q399.** Which tree has nodes with 0 or 2 children only?  
a) Extended binary tree b) BST  
c) AVL d) Heap

**Q400.** In hash table, collision is resolved by  
a) Chaining b) Recursion  
c) Tree d) Searching

**Q401.** In merge sort, the array is divided into two halves and then:  
a) Both halves are sorted recursively  
b) Sorted separately using bubble sort  
c) Sorted using selection sort  
d) Sorted using insertion sort

**Q402.** Which of the following data structure is used in Depth First Search (DFS)?  
a) Queue b) Stack c) Linked List d) Heap

**Q403.** Which of the following sorting techniques is best for linked list?  
a) Bubble Sort b) Merge Sort c) Quick Sort d) Heap Sort

**Q404.** What is the minimum number of stacks required to implement a queue?  
a) One b) Two c) Three d) Four

**Q405.** The worst-case time complexity of quick sort is:  
a) O(n²) b) O(log n) c) O(n log n) d) O(n)

**Q406.** What is the time complexity of accessing an element in an array?  
a) O(n) b) O(log n) c) O(n log n) d) O(1)

**Q407.** Which is not a tree traversal technique?  
a) Inorder b) Preorder c) Postorder d) Roundorder

**Q408.** Which of the following data structures is suitable for implementing recursion?  
a) Queue b) Stack c) Tree d) Array

**Q409.** In a graph, a cycle is a path that starts and ends at:  
a) Same vertex b) Different vertex  
c) Source d) Sink

**Q410.** In a binary tree, a node with no child is called:  
a) Internal node b) Leaf node  
c) Root node d) Subnode

**Q411.** Which sorting algorithm has best time complexity of O(n)?  
a) Merge Sort b) Heap Sort c) Counting Sort d) Quick Sort

**Q412.** Which is true about circular queue?  
a) It is a FIFO structure  
b) It overcomes linear queue overflow  
c) Rear points to the front in full condition  
d) All of the above

**Q413.** What is the height of a complete binary tree with n nodes?  
a) log(n) b) log(n+1) c) log₂(n) d) n/2

**Q414.** What is the auxiliary space complexity of merge sort?  
a) O(1) b) O(log n) c) O(n) d) O(n²)

**Q415.** Hashing handles collisions using:  
a) Probing b) Chaining  
c) Resizing d) Both a & b

**Q416.** Number of edges in a complete undirected graph with n vertices:  
a) n(n+1)/2 b) n(n–1)/2 c) n² d) n(n–1)

**Q417.** Stack is also known as:  
a) LIFO b) FIFO c) FILO d) LILO

**Q418.** Which algorithm is used to find shortest path?  
a) Dijkstra b) Kruskal  
c) Prim d) Floyd

**Q419.** What is the worst-case time complexity of insertion sort?  
a) O(n) b) O(n log n) c) O(n²) d) O(1)

**Q420.** AVL tree is a type of:  
a) Binary tree b) Balanced BST  
c) Heap d) Threaded tree

**Q421.** Which notation describes average case complexity?  
a) O(n) b) Ω(n) c) Θ(n) d) None

**Q422.** Which of the following is true for heap sort?  
a) In-place b) Stable  
c) Not recursive d) None

**Q423.** Which of the following is not a stable sort?  
a) Bubble b) Selection  
c) Insertion d) Merge

**Q424.** Which of the following is a divide and conquer algorithm?  
a) Merge sort b) Quick sort  
c) Binary search d) All

**Q425.** Which one is not a linear data structure?  
a) Stack b) Queue  
c) Array d) Graph

**Q426.** What is the primary use of a queue?  
a) Function call b) Buffer handling  
c) Recursion d) Priority queue

**Q427.** A tree with height h has how many minimum nodes?  
a) h b) h+1 c) 2h – 1 d) h – 1

**Q428.** Minimum number of queues to implement a stack:  
a) 1 b) 2 c) 3 d) 4

**Q429.** Adjacency matrix is used to represent:  
a) Graph b) Tree c) Stack d) Queue

**Q430.** Which traversal follows Left → Root → Right?  
a) Preorder b) Postorder  
c) Inorder d) Level order

**Q431.** Tree traversal technique using recursion:  
a) Inorder b) Postorder  
c) Preorder d) All

**Q432.** Data structure ideal for parsing expressions:  
a) Tree b) Stack c) Queue d) Graph

**Q433.** Postorder traversal visits:  
a) Root → Left → Right  
b) Left → Right → Root  
c) Left → Root → Right  
d) Right → Root → Left

**Q434.** Which sorting algorithm is fastest on average?  
a) Quick sort b) Merge sort  
c) Bubble sort d) Insertion sort

**Q435.** In DFS, which data structure is used?  
a) Stack b) Queue  
c) Tree d) Heap

**Q436.** Complete binary tree with 15 nodes has height:  
a) 4 b) 5 c) 3 d) 2

**Q437.** Hash table performance depends on:  
a) Load factor b) Hash function  
c) Collision handling d) All of these

**Q438.** In a min heap, the smallest element is at:  
a) Leaf b) Middle  
c) Root d) Last

**Q439.** What is the average time complexity of binary search?  
a) O(n) b) O(n log n)  
c) O(log n) d) O(1)

**Q440.** Best case for bubble sort occurs when:  
a) Data is reverse sorted  
b) Data is sorted  
c) All elements are equal  
d) None

**Q441.** Queue insertion is done at:  
a) Rear b) Front  
c) Middle d) Root

**Q442.** Which is not a characteristic of stack?  
a) LIFO b) Recursive utility  
c) Push/Pop operations  
d) Random access

**Q443.** In which tree every internal node has 2 children?  
a) AVL tree b) BST c) Complete binary tree d) Full binary tree

**Q444.** Stack overflow happens when:  
a) Too many elements pushed  
b) Too few elements  
c) Memory leak  
d) Queue full

**Q445.** Queue underflow occurs when:  
a) Removing from empty queue  
b) Adding to full queue  
c) Overflow in memory  
d) None

**Q446.** What is top in an empty stack?  
a) –1 b) 0 c) NULL d) Depends on implementation

**Q447.** Number of edges in a tree with n nodes:  
a) n b) n – 1 c) n + 1 d) 2n

**Q448.** Which of the following is used for priority scheduling?  
a) Stack b) Queue  
c) Priority queue d) Tree

**Q449.** Which technique ensures BST balance?  
a) Red-black tree  
b) AVL tree  
c) Both a & b  
d) None

**Q450.** What’s the time complexity of heapify operation?  
a) O(1) b) O(log n) c) O(n) d) O(n log n)

**Q451.** Which of the following statements is true regarding queues?  
a) FIFO b) LIFO c) FILO d) None

**Q452.** The prefix form of A – B + C is  
a) – + A B C b) + – A B C c) – A + B C d) + A – B C

**Q453.** The inorder traversal of BST results in  
a) Sorted order b) Reverse order c) Level order d) Random order

**Q454.** Which of the following operations is not possible in an array?  
a) Insert b) Delete c) Modify d) Random access

**Q455.** The number of edges in a complete graph with n vertices is  
a) n(n – 1)/2 b) n² c) n – 1 d) n

**Q456.** Which of the following is not a linear data structure?  
a) Array b) Stack c) Binary Tree d) Queue

**Q457.** Which operation is used to delete the top of stack?  
a) Delete b) Remove c) Pop d) Push

**Q458.** The postfix of (A + B) \* (C – D) is  
a) AB+CD-\* b) AB+*CD– c) A+B*C–D d) AB+CD–\*

**Q459.** Which data structure is used for implementing recursion?  
a) Queue b) Linked List c) Stack d) Tree

**Q460.** Which of the following data structures is best suited for priority scheduling?  
a) Stack b) Queue c) Deque d) Priority Queue

**Q461.** In binary heap, insertion and deletion takes  
a) O(1) time b) O(log n) time c) O(n) time d) O(n²) time

**Q462.** Which tree has maximum height for n nodes?  
a) Balanced tree b) Skewed tree c) AVL tree d) BST

**Q463.** In a BST, the left child must contain  
a) Larger values b) Equal values c) Smaller values d) None

**Q464.** Which traversal gives expression in prefix form?  
a) Inorder b) Preorder c) Postorder d) Level order

**Q465.** Which data structure is most efficient for implementing priority queue?  
a) BST b) Array c) Heap d) Stack

**Q466.** The time complexity for searching in hash table is  
a) O(log n) b) O(n) c) O(1) d) O(n log n)

**Q467.** In a graph, edges with direction are called  
a) Undirected b) Directed c) Cycles d) Paths

**Q468.** What is the degree of a leaf node in binary tree?  
a) 0 b) 1 c) 2 d) Depends on tree

**Q469.** A full binary tree of height h has how many nodes?  
a) 2^h – 1 b) 2^(h + 1) – 1 c) h d) h²

**Q470.** Stack overflow occurs when  
a) Stack becomes empty  
b) Stack is full and push is done  
c) Stack underflows  
d) None

**Q471.** An algorithm is a set of  
a) Rules b) Statements  
c) Instructions d) Data structures

**Q472.** Which of the following algorithms is not recursive?  
a) DFS b) Inorder Traversal  
c) Bubble Sort d) Factorial

**Q473.** Which of these sorting algorithms is not comparison-based?  
a) Heap Sort b) Radix Sort c) Merge Sort d) Bubble Sort

**Q474.** The best case for quicksort is  
a) Sorted data  
b) Reverse sorted  
c) Random order  
d) Equal keys

**Q475.** Which is a non-primitive data structure?  
a) Integer b) Float c) Array d) Character

**Q476.** Preorder traversal of binary tree:  
a) Root-Left-Right b) Left-Root-Right  
c) Left-Right-Root d) Right-Left-Root

**Q477.** Postorder traversal of binary tree:  
a) Root-Left-Right b) Left-Root-Right  
c) Left-Right-Root d) Right-Left-Root

**Q478.** Breadth First Search (BFS) uses  
a) Stack b) Queue c) Tree d) Graph

**Q479.** A tree is a  
a) Non-linear data structure  
b) Linear data structure  
c) Primitive structure  
d) Ordered structure

**Q480.** What is a complete binary tree?  
a) All levels are completely filled  
b) All internal nodes have two children  
c) Last level may be incomplete but all nodes are as left as possible  
d) None

**Q481.** Hash collision means  
a) Two data with same key  
b) Same data in two locations  
c) Two keys map to same index  
d) None

**Q482.** Stack follows  
a) FIFO b) LIFO c) FILO d) LILO

**Q483.** Queue follows  
a) LIFO b) FIFO c) FILO d) Random

**Q484.** Time complexity of inserting an element in heap is  
a) O(1) b) O(log n) c) O(n) d) O(n log n)

**Q485.** A graph with all edges directed is  
a) Undirected b) Directed c) Tree d) DAG

**Q486.** Minimum spanning tree can be found using  
a) BFS b) DFS c) Dijkstra d) Kruskal

**Q487.** A tree with n nodes has how many edges?  
a) n b) n – 1 c) n + 1 d) 2n

**Q488.** Level-order traversal uses  
a) Stack b) Queue c) Recursion d) Inorder

**Q489.** Binary tree can be used to implement  
a) Expression tree b) Heap  
c) BST d) All of the above

**Q490.** Degree of a node in a tree is  
a) Number of children  
b) Number of edges  
c) Number of leaves  
d) None

**Q491.** Height of a leaf node is  
a) 0 b) 1 c) 2 d) Depends

**Q492.** Recursion uses which data structure?  
a) Queue b) Stack c) Array d) Heap

**Q493.** Queue underflow happens when  
a) Deleting from empty queue  
b) Inserting in full queue  
c) Deleting from full queue  
d) None

**Q494.** Circular queue is better than linear queue because  
a) No space is wasted  
b) Faster  
c) More memory  
d) None

**Q495.** A tree with only left children is  
a) Left skewed tree  
b) Right skewed tree  
c) AVL tree  
d) Binary search tree

**Q496.** Priority Queue is implemented using  
a) Heap b) Stack c) Queue d) Tree

**Q497.** Hashing helps in  
a) Sorting b) Searching c) Insertion d) Merging

**Q498.** Dynamic data structure example  
a) Array b) Linked list  
c) Structure d) None

**Q499.** Which sorting algorithm is stable?  
a) Quick sort b) Merge sort  
c) Heap sort d) Selection sort

**Q500.** The worst-case complexity of binary search  
a) O(1) b) O(log n) c) O(n) d) O(n log n)