INTERVIEW SCHOOL

Note:-Check the Answers on Last page……

[QUESTIONS 1 TO 141] [PAGE 1 TO 14]

MULTIPLE CHOICE OBJECTIVE TYPE QUESTION

Q1. Which out of these is a non-linear data-structure:

a. arrays b.linked-lists

c. queues d. tree

Q2. A stack is a data-structure in which elements are stored and retrieved by:

a. FIFO method b.LIFO method

c. FCFS method d. None of the above

Q3. The different types of arrays are:

a. One & Multi-dimensional b. int and float

c. int,char,float d. One & Two dimensional

Q4. An array is passed into a function:

a. by value b. by reference

c. element by element d. Any of the above

Q5. A queue is a data-structure in which elements are stored and retrieved by:

a. FIFO method b.LIFO method

c. FCFS method d. None of the above

Q6. If an array with the name, A exists which of the following statements is incorrect:

a. A++ b. printf("%d",*(A+1))

c. printf("%u",A+1) d. All are correct

Q7. An uninitialized pointer is known as:

a. dangling pointer b. NULL pointer

c. generic pointer d. None of the above

Q8. The unary operator used with pointer variable to indirectly access the contents of memory location pointed to by the pointer is called

a. Address-of operator b. dot operator

c. indirection operator d. asterisk operator

Q9. The arithmetic operation performed on pointer variables:

a. multiplication of an integer with pointer b. addition of two pointers

c. subtraction of two pointers d. addition of a float to a pointer

Q10. Two ways to access elements of an array are:

a. by value and by reference b. indexed and pointer notation

c. sequential and random d. none of the above

Q11. The functions used for dynamic memory allocation are:

a. delete and free b. free and realloc

c. malloc and free d. malloc and calloc

Q12. An array can be categorized as which type of data-structure:

a. dynamic
b. static
c. int/char/float
d. multi-dimensional

Q13. The function used in C to de-allocate a memory block is:

a. delete
b. free
c. malloc
d. realloc

Q14. A collection of elements of different data-types is known as:

a. array
b. union
c. structure
d. linked list

Q15. To access elements of a structure through a pointer , the operator used is:

a. .
b. ->
c. <-
d. *

Q16. A structure having atleast one of its members of the same type as the structure itself:

a. reference structure
b. nested structure
c. self-referential structure
d. none of the above

Q17. A one-way list is called:

a. circular linked list
b. array
c. queue
d. single linked list

Q18. An orphaned block is the result of:

a. memory leak
b. garbage collection
c. free function
d. all of the above

Q19. A ____ pointer can point to any data-type:

a. NULL pointer
b. void pointer
c. dangling pointer
d. such a pointer does not exist

Q20. If p1 and p2 are pointers of integer type and x is also an integer-type of variable, which of these expressions is valid:

a. p1*x
b. p2+x
c. p1+p2
d.p1*p2

Q21. A linked list can be represented using two ways which are:

a. single and double
b. single and circular
c. static and dynamic
d. any of the above

Q22. Dynamic memory allocation for a node in a linked list is done from:

a. RAM
b. ROM
c. Hard disk
d. Heap

Q23. A linked list is a linear collection of homogeneous elements called:

a. structures
b. nodes
c. data
d. none of the above

Q24. Insertion in a linked list can be done from:

a. beginning

b. end

c. middle

d. all of the above

Q25. The traversal directions possible in a double-linked list are:

a. forward

b. backward

c. forward and backward

d. right

Q26. A node in a double linked list comprises of:

a: information field

b: information field and next pointer

c: information field, next pointer and thread field

d: information field, next pointer, previous pointer & previous pointer

Q27. The situation in which the user tries to delete a node from an empty linked list is called:

a: empty

b: free

c. overflow

d. underflow

Q28. When a new node is inserted in between a linked list, which of these is true:

a: only the nodes appearing after the new node needs to be moved

b: only the nodes appearing before the new node needs to be moved

c: the nodes appearing before and after the new node need to be moved

d: None of the above

Q29. The situation in which memory is not available for the allocation of a new node:

a: empty

b: free

c. overflow

d. underflow

Q30. A linear linked list in which the next field of the last node points back to the first node is termed as:

a: single linked list

b: double linked list

c: circular linked list

d: reversed linked list

Q31. A new node can be dynamically inserted anytime in a linked list, for which the memory manager maintains a special list known as

a: available list

b: free-storage list

c: single linked list

d: dynamic storage list

Q32. The insertion/deletion operations on a stack are respectively known as:

a: insert and delete

b: enter and exit

c: push and pop

d: none of the above

Q33. A stack could be implemented using:

a. single and double

b. single and circular

c: array and linked list

d: any of the above

Q34. The pointer used to point to the element in the beginning of the stack is called:

a: start

b: front

c: root

d: top

Q35. Postfix notation is also know as:

a: polish notation
b: reverse polish notation

c: post notation
d: post-operator notation

Q36. A linked list is which type of data-structure:

a: static
b: non-linear

c: linear
d: none of the above

Q37. Prefix notation is also known as:

a: polish notation
b: reverse polish notation

c: pre notation
d: post-operator notation

Q38. When converting an infix expression to postfix using algorithm, when '(' is encountered, it is:

a: added to postfix string
b: pushed and operators are popped from the stack

c: pushed onto the stack
d: options a and c are performed

Q39. The data-structure in which both insertion and deletion take place from the beginning:

a: linked list
b: queue

c: tree
d: stack

Q40. The postfix expression: 5 6 2 + * 12 4 /- when evaluated gives the following result:

a: 37
b: -37

c: 40
d: 3

Q41. The condition top=-1 indicates that:

a: stack has only one element
b: stack is full

c: stack is empty
d: none of these

Q42. The validity of an expression containing nested parentheses could be checked using:

a: linked list
b: queue

c: tree
d: stack

Q43. A string could be reversed using the data-structure:

a: linked list
b: queue

c: tree
d: stack

Q44. The notation in which the operator occurs between the operands is called:

a: infix notation
b: prefix notation

c: postfix notation
d: post-operator notation

Q45. In a normal queue, the underflow situation occurs when:

a: rear=max -1
b: front= -1

c: rear=front
d: rear=size-1

Q46. In a circular queue, one of the situations when overflow occurs:

a: rear=size-1
b: rear=front

c: (rear+1)=front
d: front=-1

Q47. In postfix expression, the operator is placed:

a: in-between the operands b: after the operands

c: before the operands d: none of these

Q48. In a priority queue, the elements with the same priority are processed according to:

a: First come first served basis b: priority

c: last in first come basis d: no specific order

Q49. The deque in which insertion is done at one end and deletion from both ends:

a: input-restricted deque b: output-restricted deque

c: input-output restricted deque d: Any of the above

Q50. A circular array queue with space for 10 elements in which front =6 and rear=9, insertion of next element will take place at position:

a: 0 b: 5

c: 7 d: insertion can not take place due to overflow situation

Q51. A data-structure in which each element is assigned a priority and the elements are added/removed according to that priority :

a: priority list b: priority queue

c: stack d: none of the above

Q52. A linear data-structure in which elements could be inserted/deleted at either end but not in the middle:

a: queue b: stack

c: deque d: circular queue

Q53. In a circular queue with 10 elements, if front is at 9 and rear at 4, the deletion of an element will make front point to which position:

a: 0 b: -1

c: 3 d: 5

Q54. A non-linear hierarchical type of data-structure:

a: graph b: tree

c: array d: deque

Q55. All leaf nodes of a tree are termed as:

a: terminal nodes b: non-terminal nodes

c: child nodes d: internal nodes

Q56. The root node is

a: terminal node b: internal nodes

c: child node d: none of the above

Q57. The nodes belonging to the same parent are known as:

a: descendants b: external nodes

c: child nodes d: siblings

Q58. A tree in which the degree of each node is either 0 or 2:

a: complete binary tree
b: binary search tree
c: strictly binary tree
d: none of the above

Q59. A binary tree in which all the leaf nodes of the tree are at the same level:

a: complete binary tree
b: binary search tree
c: strictly binary tree
d: none of the above

Q60. At any level x of a binary tree, the maximum number of nodes are:

a. 2^x
b: 2*x
c: 2+x
d: none of the above

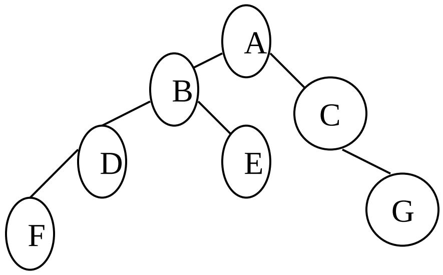
Q61. What is the root node for the algebraic expression : a*(b+c)-d, if it is represented in the form of a tree:

a: -
b: +
c: a
d: *

Q62. In inorder traversal of a binary tree, the root node is visited:

a: after the traversal of right & left subtrees
b: before the traversal of right and left subtrees
c. in-between the traversal of left and right subtrees
d: none of these

Q63. The height of the following binary tree is:



a: 3
b: 4
c: 5
d: 2

Q64. A binary search tree is also known as:

a: B-tree
b: binary sorted tree
c. binary ordered tree
d: B+ tree

Q65. A binary tree in which the node-values are not repeated is called:

a: B-tree
b: binary search tree
c. binary ordered tree
d: B+ tree

Q66. A binary search tree in which the nodes have been inserted in the following order:60,55,95,40,30,100,35, the node with the value 47 will be inserted to the:

a: right of node with value 40
b: right of node with value 55
c: right of node with value 35
d: left of node with value 30

Q67. In the following post-order traversal of a binary tree: E,C,K,A,H,B,G,D,F, the root node is:

a: E
b: H
c: F
d: D

Q68. The complexity of bubble-sort algorithm is:

a: O(n^2)
b: O(n)
c: O(log n)
d: O(n log n)

Q69. Binary search is more suitable for:

a: array
b: linked list
c: stack
d: any of the above

Q70. The complexity of binary-search algorithm is:

a: O(log n)
b: O(n log n)
c: O(n)
d: O(n^2)

Q71. The calloc() function can be used to allocate:

a: multiple blocks of memory
b: single block of memory
c: two blocks of memory
d: none of these

Q72. The postfix expression of the infix expression: A+B*(C+D)/F+D*E is:

a: AB+CD+*F/D+E*
b: ABCD+*F/+DE*+
c: A*B+CD/F*DE++
d: A+*BCD/F*DE++

Q73. A linear list of elements in which deletion can be done from one end and insertion can take place at the other end is called:

a: queue
b: stack
c: tree
d: branch

Q74. Which data-structure is needed to convert infix notation to postfix notation:

a: queue
b: stack
c: tree
d: linked list

Q75. Which of the following sorting procedures is the slowest:

a: Quick sort
b: bubble sort
c: Shell sort
d: insertion sort

Q76. The 'C' declaration: int b[100]; reserves ____ successive memory locations, each large enough to contain a single integer:

a: 200
b: 10000
c: 10
d: 100

Q77. If n elements are to be sorted, the complexity of selection-sort is:

a: O(1)
b: O(log n)
c: O(n)
d: O(n^2)

Q78. The operation of processing each element in the list is known as:

a: sorting
b: merging

c: inserting d: traversal

Q79. Arrays are best data structures:

a: for relatively permanent collections b: when the size of structure is constantly changing

c: for both the above situations d: for none of the above situations

Q80. The elements of an array are stored successively in memory cells because:

a: in this way the computer can calculate the address of other elements keeping track of address of first element

b: computer architecture allows arrays to be stored serially only

c: both of the above d: none of the above

Q81. Pick the odd one out:

a: insertion sort b: selection sort

c: counting sort d: merge sort

Q82. If you wanted to make sure that closing parentheses ‘)’ match the opening parentheses ‘(‘ in a mathematical expression, which data-structure could help you?

a: hash table b: stack

c: queue d: tree

Q83. The estimated amount of time required in executing an algorithm is referred to as _____ of the algorithm.

a: time complexity b: space complexity

c: time and space complexity d: none of the above

Q84. If all the data to be sorted does not fit entirely in main memory, the sorting technique used is:

a: internal sorting b: external sorting

c: merge sorting d: sorting can not be performed

Q85. The searching technique suitable for unsorted arrays:

a: binary search b: linear search

c: any of these d: none of these

Q86. A theoretical measure of algorithm execution, usually the time/ memory needed , given the problem size n , is referred to as:

a: Big O notation b: Polish notation

c: Time notation d: space complexity

Q87. The technique of collecting unused memory is known as:

a: garbage collection b: Dynamic memory allocation

c: static memory allocation d: none of these

Q88. The root node of a binary tree whose preorder traversal is: F,B,A,D,C,E,G, I, H is:

a: F b: H

c: C d: none of these

Q89. The post-order traversal of an arithmetic expression will result in the expression being represented as:

a: postfix b: prefix
c: infix d: none of the above

Q90. The main measures for the efficiency of an algorithm are:

a: processor and memory b: complexity and capacity
c: time and space d: data and space

Q91. Which of the following cases does not exist in complexity theory:

a: best case b: worst case
c: average case d: Null case

Q92. The worst case occurs in linear search algorithm when:

a: item is in the middle of the array b: item is not in the array
c: item is the last element in the array d: item is the last element in the array or not in the array at-all

Q93. The complexity of merge sort algorithm is:

a: O(n) b: O(log n)
c: $O(n^2)$ d: O(n log n)

Q94. The complexity of linear search algorithm is:

a: O(n) b: O(log n)
c: $O(n^2)$ d: O(n log n)

Q95. Which of the following data structures is not a linear data structure:

a: arrays b: linked lists
c: both of the above d: none of the above

Q96. Linked lists are best suited:

a: for relatively permanent collections b: when the size of structure is constantly changing
c: for both the above situations d: for none of the above situations

Q97. The memory address of the first element of an array is called:

a: floor address b: foundation address
c: first address d: base address

Q98. The memory address of the fifth element of an array can be calculated by the formula:

a: Base(Array)+w(5-lower bound) where w is the size of each element of array b: Base(Array[5])+(5-lower bound)
c: Base(Array[5])+(5-upper bound) d: none of the above

Q99. Which of the following data-structures are indexed structures:

a: linear arrays b: linked lists
c: both of the above d: none of the above

Q100. Which of the following is not the required condition for binary search algorithm:

a: the list must be sorted
b: a direct access to middle element is needed
c: a mechanism to delete/insert elements in list
d: None of the above

Q101. Which of the following data structures can't store non-homogeneous data-elements:

a: Arrays
b: Records
c: Pointers
d: None

Q102. Which of the following statements is false:

a: Arrays are static data structures
b: data elements in linked list need not be stored in adjacent space in memory
c: pointer stores the next data element of a list
d: linked lists are collection of nodes that contain information part & next pointer

Q103. Which of the following is a two-way list:

a: grounded header list
b: circular header list
c: linked list with header & trailer nodes
d: none of the above

Q104. The terms "push" and "pop" are related to:

a: array
b: lists
c: stacks
d: all of the above

Q105. The depth of a complete binary tree is given by:

a: n log n
b: n log n +1
c: log n
d: log n +1

Q106. When representing any algebraic expression E which uses only binary operations in a 2-tree:

a: the variables in E will appear as external nodes and operations as internal nodes
b: the operations in E will appear as external nodes and variables as internal nodes
c: the variables and operations in E will appear only as internal nodes
d: the variables and operations in E appear only as external nodes

Q107. An algorithm that calls itself directly or indirectly is known as:

a: sub-algorithm
b: recursive algorithm
c: polish notation
d: traversal algorithm

Q108. The inorder traversal of tree will yield a sorted listing of elements of tree:

a: binary tree
b: binary search tree
c: heaps
d: none of the above

Q109. Value of first linked list index is:

a: 1
b: 0
c: -1
d: none of these

Q110. A ________ is a data-structure that organizes data similar to a line in the super-market, where the first one in the line is the first to be out:

a: queue
b: stacks
c: any of the two above
c: none of these

Q111. Which of the following abstract data types is not used by integer abstract data type group?

a: short
b: int
c: float
d: long

Q112. In a heap tree:

a: value in a node is greater than every value in left subtree and smaller than right subtree
b: value in a node is greater than the value of its child nodes
c: both the above conditions
d: none of the above conditions

Q113. The variables which can be accessed by all modules in a program, are known as:

a: local variables
b: internal variables
c: external variables
d: global variables

Q114. The post order traversal of a binary tree is :DEBFCA, find out the preorder traversal:

a: ABFCDE
b: ADBFEC
c: ABDECF
d: ABDCEF

Q115. Which of the following algorithms is of divide and conquer type:

a: bubble-sort
b: insertion sort
c: quick sort
d: all of the above

Q116. One of the applications of a linked list:

a: Polynomial evaluation
b: Postfix expression evaluation
c: determining the distance traveled
d: none of these

Q117. A tree having any number of nodes:

a: binary tree
b: general tree
c: B-tree
d: AVL tree

Q118. A set of several trees that are not linked to each other in any way

a: Forest
b: Graphs
c: B-trees
d: none of these

Q119. All the non-leaf nodes except the root node in a multi-way search tree of order, n have atleast:

a: n-1 children
b: n children
c: n/2 children
d: n*2 children

Q120. Heaps are of two types:

a: high and low
b: max and min
c: B and B+
d: none of the above

Q121. Incase of min-heap, the value present in any node is:

a: greater than all its children
b: smaller than all its children

c: equal to all its children d: greater than values in left subtree and smaller than values in right subtree

Q122. A min-heap is also known as:

a: decreasing heap b: descending heap
c: low heap d: none of these

Q123. A max-heap is also known as:

a: increasing heap b: ascending heap
c: high heap d: none of these

Q124. A tree in which the value in every node is more than node-values in its left subtree and less than node-values in its right subtree:

a: binary sorted tree b: B-tree
c: B+ tree d: AVL tree

Q125. A matrix which has most of its values equal to 0:

a: sparse matrix b: zero-matrix
c: empty matrix d: none of the above

Q126. A sparse matrix can also be represented using:

a: queue b: stack
c: tree d: linked list

Q127. A B-tree grows at the:

a: root b: leaves
c: braches d: any of the above

Q128. A binary tree grows at the

a: root b: leaves
c: braches d: any of the above

Q129. Shell sort is an improvisation over:

a: quick- sort b: merge-sort
c: insertion-sort d: none of these

Q130. To reduce disk-accesses while searching for a record, the tree used is:

a: binary sorted tree b: B-tree
c: general tree d: AVL tree

Q131. While calculating time-complexity, the program-time which is considered is:

a: compile time b: execution time
c: both compile and run-time d: none of the above

Q132. The time complexity of the following algorithm is:
sum(a,n){ s=0; for i= 1 to n{s=s+a[i]; } return s;}

a: 3n+2 b: 2n +3
c: n+1 d: 2n+2

Q133. Complexity of heap sort

a: O(n) b: O(log n)

c: O(n^2) d: O(n log n)

Q134. If there are more than one paths between two nodes, it is a:

a: tree b: graph

c: circular linked list d: none of the above

Q135. The leaf nodes of a tree have height equal to:

a: height of the tree b: zero

c: one d: none of these

Q136. A binary tree with n internal nodes has a max. of external nodes equal to:

a: n-1 b: n+1

c: n d: n/2

Q137. Height of a full binary tree with n internal nodes is:

a: n log n b: n

c: n+1 d: log n

Q138. The degree of a leaf node is:

a: 1 b: 0

c: -1 d:2

Q139. A right in-threaded binary tree contains:

a: inorder successor b: inorder predecessor

c: NULL d: preorder successor

Q140. The algorithm used in dynamic memory allocation with minimum time:

a: First fit b: Best fit

c: Worst fit d: Next fit

Q141. The algorithm used in dynamic memory allocation which results in minimum fragmentairst fit

b: Best fit

c: Worst fit d: Next fit

Answers

1(d)	2(b)	3(c)	4(b)	5(a)	6(a)	7(a)	8(c)
9(c)	10(b)	11(d)	12(b)	13(b)	14(c)	15(b)	16(c)
17(d)	18(a)	19(b)	20(b)	21(c)	22(d)	23(b)	24(d)
25(c)	26(c)	27(d)	28(d)	29(c)	30(c)	31(b)	32(c)
33(c)	34(d)	35(b)	36(c)	37(a)	38(c)	39(d)	40(a)
41(c)	42(d)	43(d)	44(a)	45(b)	46(c)	47(b)	48(a)
49(a)	50(a)	51(b)	52(c)	53(a)	54(b)	55(a)	56(b)
57(d)	58(c)	59(a)	60(a)	61(a)	62(c)	63(a)	64(b)
65(b)	66(a)	67(c)	68(a)	69(a)	70(a)	71(a)	72(b)
73(a)	74(b)	75(b)	76(d)	77(d)	78(d)	79(a)	80(a)
81(c)	82(b)	83(a)	84(b)	85(b)	86(a)	87(a)	88(a)
89(a)	90(c)	91(d)	92(d)	93(d)	94(a)	95(d)	96(b)
97(d)	98(a)	99(a)	100(c)	101(a)	102(c)	103(d)	104(c)
105(d)	106(a)	107(b)	108(b)	109(b)	110(a)	111(c)	112(b)
113(d)	114(c)	115(c)	116(a)	117(b)	118(a)	119(c)	120(b)
121(b)	122(b)	123(b)	124(a)	125(a)	126(d)	127(a)	128(b)
129(c)	130(b)	131(b)	132(b)	133(d)	134(b)	135(b)	136(b)
137(d)	138(b)	139(a)	140(a)	141(b)			