

Data Structures Interview Questions :-

1. What is data structure?

a data structure is a way of organizing data that considers not only the items stored, but also their relationship to each other. advance knowledge about the relationship between data items allows designing of efficient algorithms for the manipulation of data.

2. List out the areas in which data structures are applied extensively?

- compiler design,
- operating system,
- database management system,
- statistical analysis package,
- numerical analysis,
- graphics,
- artificial Intelligence,
- simulation

3. What are the major data structures used in the following areas : Rdbms, network data model and Hierarchical data model.

Rdbms – array (i.e. array of structures)
network data model = graph
Hierarchical data model = trees

4. If you are using c language to implement the heterogeneous linked list, What pointer type will you use?

the heterogeneous linked list contains different data types in its nodes and we need a link, pointer to connect them. It is not possible to use ordinary pointers for this. so we go for void pointer. Void pointer is capable of storing pointer to any type as it is a generic pointer type.

5. Minimum number of queues needed to implement the priority queue?

two. one queue is used for actual storing of data and another for storing priorities.

6. What is the data structures used to perform recursion?

stack. because of its Lifo (last In first out) property it remembers its 'caller' so knows whom to return when the function has to return. Recursion makes use of system stack for storing the return addresses of the function calls.

every recursive function has its equivalent iterative (non-recursive) function. even when such equivalent iterative procedures are written, explicit stack is to be used.

7. What are the notations used in evaluation of arithmetic expressions using prefix and postfix forms?

polish and Reverse polish notations.

8. convert the expression ((a + b) * c – (d – e) ^ (f + g)) to equivalent prefix and postfix notations.

prefix notation: – * +abc ^ – de + fg
postfix notation: ab + c * de – fg + ^ –

9. sorting is not possible by using which of the following methods? (Insertion, selection, exchange, deletion)

sorting is not possible in deletion, using insertion we can perform insertion sort, using selection we can perform selection sort, using exchange we can perform the bubble sort (and other similar sorting methods), but no sorting method can be done just using deletion.

10. What are the methods available in storing sequential files ?

1. straight merging,
2. natural merging,
3. polyphase sort,
4. distribution of Initial runs.

11. list out few of the application of tree data-structure?

- the manipulation of arithmetic expression,
- symbol table construction,
- syntax analysis.

12. list out few of the applications that make use of multilinked structures?

- sparse matrix,
- Index generation.

13. In tree construction which is the suitable efficient data structure? (array, linked list, stack, queue)

linked list is the suitable efficient data structure.

14. What is the type of the algorithm used in solving the 8 queens problem?

backtracking.

15. In an aVl tree, at What condition the balancing is to be done?

If the 'pivotal value' (or the 'Height factor') is greater than 1 or less than -1.

16. What is the bucket size, when the overlapping and collision occur at same time?

one. If there is only one entry possible in the bucket, when the collision occurs, there is no way to accommodate the colliding value. this results in the overlapping of values.

17. classify the Hashing functions based on the various methods by which the key value is found.

- direct method,
- subtraction method,
- modulo-division method,
- digit-extraction method,
- mid-square method,
- folding method,
- pseudo-random method.

18. What are the types of collision Resolution techniques and the methods used in each of the type?

open addressing (closed hashing), the methods used include: overflow block.
closed addressing (open hashing), the methods used include: linked list, binary tree.

19. In Rdbms, What is the efficient data structure used in the internal storage representation?

b+ tree. because in b+ tree, all the data is stored only in leaf nodes, that makes searching easier. this corresponds to the records that shall be stored in leaf nodes.

20. What is a spanning tree?

a spanning tree is a tree associated with a network. all the nodes of the graph appear on the tree once. a minimum spanning tree is a spanning tree organized so that the total edge weight between nodes is minimized.

21. does the minimum spanning tree of a graph give the shortest distance between any 2 specified nodes?

no. the minimal spanning tree assures that the total weight of the tree is kept at its minimum. but it doesn't mean that the distance between any two nodes involved in the minimum-spanning tree is minimum.

22. Which is the simplest file structure? (sequential, Indexed, Random)

sequential is the simplest file structure.

23. Whether linked list is linear or non-linear data structure?

according to access strategies linked list is a linear one.
according to storage linked list is a non-linear one.

24. define double linked list?

It is a collection of data elements called nodes,

where each node is divided into three parts:

- an info field that contains the information stored in the node.
- left field that contain pointer to node on left side.
- Right field that contain pointer to node on right side.
- adv Java Interview questions

25. Is It necessary to sort a file before searching a particular Item ?

1. If less work is involved in searching a element than to sort and then extract, then we don't go for sort.
2. If frequent use of the file is required for the purpose of retrieving specific element, it is more efficient to sort the file.
3. thus it depends on situation.

26. What are the Issues that Hamper the efficiency In sorting a file?

the issues are:

- length of time required by the programmer in coding a particular sorting program.
- amount of machine time necessary for running the particular program.
- the amount of space necessary for the particular program .
- object oriented analysis and design Interview questions

27. calculate the efficiency of sequential search?

the number of comparisons depends on where the record with the argument key appears in the table.

- If it appears at first position then one comparison
- If it appears at last position then n comparisons
- average=(n+1)/2 comparisons
- unsuccessful search n comparisons
- number of comparisons in any case is o (n).

28. Is any Implicit arguments are passed to a function When It is called?

yes there is a set of implicit arguments that contain information necessary for the function to execute and return correctly, one of them is return address which is stored within the function's data area, at the time of returning to calling program the address is retrieved and the function branches to that location.

29. parenthesis Is never Required In postfix or prefix expressions, Why?

parenthesis is not required because the order of the operators in the postfix /prefix expressions determines the actual order of operations in evaluating the expression.

30. list out the areas In Which data structures are applied extensively?

- compiler design,
- operating system,
- database management system,
- statistical analysis package,
- numerical analysis,
- graphics,
- artificial Intelligence,
- simulation.

31. What are the major data structures used In the following areas : network data model & Hierarchical data model?

Rdbms – array (i.e. array of structures)
network data model – graph
Hierarchical data model – trees

32. If you are using c language to Implement the Heterogeneous linked list, What pointer type Will you use?

the heterogeneous linked list contains different data types in its nodes and we need a link, pointer to connect them. It is not possible to use ordinary pointers for this. so we go for void pointer. Void pointer is capable of storing pointer to any type as it is a generic pointer type.

33. minimum number of queues needed to Implement the priority queue?

two. one queue is used for actual storing of data and another for storing priorities.

34. What Is the data structures used to perform Recursion?

stack. because of its Lifo (last In first out) property it remembers its 'caller' so knows whom to return when the function has to return. Recursion makes use of system stack for storing the return addresses of the function calls.

every recursive function has its equivalent iterative (non-recursive) function. even when such equivalent iterative procedures are written, explicit stack is to be used.

35. What are the notations used In evaluation of arithmetic expressions using prefix and postfix forms?

polish and Reverse polish notations.

36. convert the expression ((a + b) * c – (d – e) ^ (f + g)) to equivalent prefix and postfix notations?

prefix notation:

^ – * +abc – de + fg

postfix notation:

ab + c * de – fg + ^

37. list out few of the application of tree data-structure?

the manipulation of arithmetic expression, symbol table construction & syntax analysis.

38. list out few of the applications that make use of multilinked structures?

sparse matrix, Index generation.

39. What Is the type of the algorithm used In solving the 8 queens problem?

backtracking.

40. In an avl tree, at What condition the balancing Is to be done?

If the 'pivotal value' (or the 'Height factor') is greater than 1 or less than -1.

41. there are 8, 15, 13, 14 nodes Were there In 4 different trees. Which of them could Have formed a full binary tree?

In general: there are 2n-1 nodes in a full binary tree. by the method of elimination:

full binary trees contain odd number of nodes. so there cannot be full binary trees with 8 or 14 nodes, so rejected. With 13 nodes you can form a complete binary tree but not a full binary tree. so the correct answer is 15.

42. In Rdbms, What Is the efficient data structure used In the Internal storage Representation?

b+ tree. because in b+ tree, all the data is stored only in leaf nodes, that makes searching easier. this corresponds to the records that shall be stored in leaf nodes.

43. What Is a spanning tree?

a spanning tree is a tree associated with a network. all the nodes of the graph appear on the tree once. a minimum spanning tree is a spanning tree organized so that the total edge weight between nodes is minimized.

44. Does the minimal spanning tree of a graph give the shortest distance between any 2 specified nodes?

no! minimal spanning tree assures that the total weight of the tree is kept at its minimum. but it doesn't mean that the distance between any two nodes involved in the minimal-spanning tree is minimum.

45. difference between calloc and malloc ?

malloc: allocate n bytes.
calloc: allocate m times n bytes initialized to 0.

46. What are the major data structures used In the following areas : Rdbms, network data model & Hierarchical data model?

- Rdbms array (i.e. array of structures)
- network data model graph
- Hierarchical data model trees.

47. Which file contains the definition of member functions?

definitions of member functions for the linked list class are contained in the linkedlist.cpp file.

48. How is any data structure application Is classified among files?

a linked list application can be organized into a header file, source file and main application file. the first file is the header file that contains the definition of the node structure and the linkedlist class definition. the second file is a source code file containing the implementation of member functions of the linkedlist class. the last file is the application file that contains code that creates and uses the linkedlist class.

49. What member function places a new node at the end of the linked list?

the appendnode() member function places a new node at the end of the linked list. the appendnode() requires an integer representing the current data of the node.

50. What Is linked list ?

linked list is one of the fundamental data structures. It consists of a sequence of ? nodes, each containing arbitrary data fields and one or two ("links") pointing to the next and/or previous nodes. a linked list is a self-referential datatype because it contains a pointer or link to another data of the same type. linked lists permit insertion and removal of nodes at any point in the list in constant time, but do not allow random access.

51. What does each entry In the linked list called?

each entry in a linked list is called a node. think of a node as an entry that has three sub entries. one sub entry contains the data, which may be one attribute or many attributes. another points to the previous node, and the last points to the next node. When you enter a new item on a linked list, you allocate the new node and then set the pointers to previous and next nodes.

52. How Is the front of the queue calculated ?

the front of the queue is calculated by front = (front+1) % size.

53. Why Is the Isempty() member method called?

the isempty() member method is called within the dequeue process to determine if there is an item in the queue to be removed i.e. isempty() is called to decide whether the queue has at least one element. this method is called by the dequeue() method before returning the front element.

54. Which process places data at the back of the queue?

enqueue is the process that places data at the back of the queue.

55. What Is the Relationship between a queue and Its underlying array?

data stored in a queue is actually stored in an array. two indexes, front and end will be used to identify the start and end of the queue.

When an element is removed front will be incremented by 1. In case it reaches past the last index available it will be reset to 0. then it will be checked with end. If it is greater than end queue is empty.

When an element is added end will be incremented by 1. In case it reaches past the last index available it will be reset to 0. after incrementing it will be checked with front. If they are equal queue is full.

56. What Is a queue ?

a queue is a sequential organization of data. a queue is a first in first out type of data structure. an element is inserted at the last position and an element is always taken out from the first position.

57. What does Isempty() member method determines?

isempty() checks if the stack has at least one element. this method is called by pop() before retrieving and returning the top element.

58. What method Removes the Value from the top of a stack?

the pop() member method removes the value from the top of a stack, which is then returned by the pop() member method to the statement that calls the pop() member method.

59. What method Is used to place a Value onto the top of a stack?

push() method, push is the direction that data is being added to the stack. push() member method places a value onto the top of a stack.

60. Run time memory allocation Is Known as ?

allocating memory at runtime is called a dynamically allocating memory. In this, you dynamically allocate memory by using the new operator when declaring the array.

for example : int grades[] = new int[10];

61. How do you assign an address to an element of a pointer array ?

We can assign a memory address to an element of a pointer array by using the address operator, which is the ampersand (&), in an assignment statement such as ptemplee[0] = &projects[2];

62. Why do We use a multidimensional array?

a multidimensional array can be useful to organize subgroups of data within an array. In addition to organizing data stored in elements of an array, a multidimensional array can store memory addresses of data in a pointer array and an array of pointers.

multidimensional arrays are used to store information in a matrix form.

e.g; a railway timetable, schedule cannot be stored as a single dimensional array. one can use a 3-d array for storing height, width and length of each room on each floor of a building.

63. What Is significance of " * " ?

the symbol "*" tells the computer that you are declaring a pointer. actually it depends on context.

In a statement like int *ptr; the "*" tells that you are declaring a pointer.
In a statement like int i = *ptr; it tells that you want to assign value pointed to by ptr to variable i.
the symbol "*" is also called as Indirection operator/ dereferencing operator.

64. Is pointer a Variable?

yes, a pointer is a variable and it can be used as an element of a structure and as an attribute of a class in some programming languages such as c++, but not Java. However, the contents of a pointer is a memory address of another location of memory, which is usually the memory address of another variable, element of a structure, or attribute of a class.

65. How many parts are there In a declaration statement?

there are two main parts, variable identifier and data type and the third type is optional which is type qualifier like signed/unsigned.

66. How memory Is Reserved using a declaration statement ?

memory is reserved using data type in the variable declaration. a programming language implementation has predefined sizes for its data types.

for example:

- in c# the declaration int i; will reserve 32 bits for variable i.
- a pointer declaration reserves memory for the address or the pointer variable, but not for the data that it will point to. the memory for the data pointed by a pointer has to be allocated at runtime.
- the memory reserved by the compiler for simple variables and for storing pointer address is allocated on the stack, while the memory allocated for pointer referenced data at runtime is allocated on the heap.

67. What Is Impact of signed numbers on the memory?

sign of the number is the first bit of the storage allocated for that number, so you get one bit less for storing the number. for example if you are storing an 8-bit number, without sign, the range is 0-255. If you decide to store sign you get 7 bits for the number plus one bit for the sign. so the range is -128 to +127.

68. What Is precision?

precision refers the accuracy of the decimal portion of a value. precision is the number of digits allowed after the decimal point.

69. What Is the difference between null and Void pointer?

null can be value for pointer type variables.
Void is a type identifier which has not size.
null and void are not same. example: void* ptr = null;

70. What Is the difference between array and stack?

stackK follows Lifo. thus the item that is first entered would be the last removed.

In array the items can be entered or removed in any order. basically each member access is done using index. no strict order is to be followed here to remove a particular element.

array may be multidimensional or one-dimensional but stack should be linear data structure.

71. tell How to check Whether a linked list Is circular ?

create two pointers, each set to the start of the list. update each as follows:

```
while (pointer1)
{
    pointer1 = pointer1->next;
    pointer2 = pointer2->next;
    if(pointer2==pointer2->next;
    if(pointer1 == pointer2)
    {
        print ("circularm");
    }
}
```

72. Whether linked list Is linear or non-linear data structure?

- according to access strategies linked list is a linear one.
- according to storage linked list is a non-linear one.

73. If you are using c language to Implement the Heterogeneous linked list, What pointer type Will you use?

the heterogeneous linked list contains different data types in its nodes and we need a link, pointer to connect them. It is not possible to use ordinary pointers for this. so we go for void pointer. Void pointer is capable of storing pointer to any type as it is a generic pointer type.

74. What Is a node class?

a node class is a class that, relies on the base class for services and implementation, provides a wider interface to users than its base class, relies primarily on virtual functions in its public interface depends on all its direct and indirect base class can be understood only in the context of the base class can be used as base for further derivation can be used to create objects. a node class is a class that has added new services or functionality beyond the services inherited from its base class.

75. When can you tell that a memory leak Will occur?

a memory leak occurs when a program loses the ability to free a block of dynamically allocated memory.

76. How many different trees are possible With=103?

1014 – for example, consider a tree with 3 nodes(n=3), it will have the maximum combination of 5 different (ie, 23 – 3 =?= 5) trees.

77. How can I search for data In a linked list?

unfortunately, the only way to search a linked list is with a linear search, because the only way a linked list's members can be accessed is sequentially. sometimes it is quicker to take the data from a linked list and store it in a different data structure so that searches can be more efficient.

78. define data structures?

data structures is defined as the way of organizing all data items that consider not only the elements stored but also stores the relationship between the elements.

79. define primary data structures?

primary data structures are the basic data structures that directly operate upon the machine instructions. all the basic constants (integers, floating-point numbers, character constants, string constants) and pointers are considered as primary data structures.

80. define static data structures?

a data structure formed when the number of data items are known in advance is referred as static data structure or fixed size data structure.

81. list some of the static data structures In c?

some of the static data structures in c are arrays, pointers, structures etc.

82. define dynamic data structures?

a data structure formed when the number of data items are not known in advance is known as dynamic data structure or variable size data structure.

83. list some of the dynamic data structures In c?

some of the dynamic data structures in c are linked lists, stacks, queues, trees etc.

84. define linear data structures?

linear data structures are data structures having a linear relationship between its adjacent elements.

eg; linked lists.

85. define non-linear data structures?

non-linear data structures are data structures that don't have a linear relationship between its adjacent elements but have a hierarchical relationship between the elements.

eg; trees and graphs.

86. define linked lists?

linked list consists of a series of structures, which are not necessarily adjacent in memory. each structure contains the element and a pointer to a structure containing its successor. We call this the next pointer, the last cell's next pointer points to null.

87. state the different types of linked lists?

the different types of linked list include singly linked list, doubly linked list and circular linked list.

88. list the basic operations carried out In a linked list?

the basic operations carried out in a linked list include:

- creation of a node.
- Insertion of a node.
- deletion of a node.
- modification of a node.
- traversal of the list.

89. list out the advantages of using a linked list?

It is not necessary to specify the number of elements in a linked list during its declaration.
linked list can grow and shrink in size depending upon the insertion and deletion that occurs in the list.
Insertions and deletions at any place in a list can be handled easily and efficiently.
a linked list does not waste any memory space.

90. list out the disadvantages of using a linked list?

searching a particular element in a list is difficult and time consuming.
a linked list will use more storage space than an array to store the same number of elements.

91. list out the applications of a linked list?

some of the important applications of linked lists are manipulation of polynomials, sparse matrices, stacks and queues.

92. state the difference between arrays and linked lists?

submit

93. define a stack?

stack is an ordered collection of elements in which insertions and deletions are restricted to one end. the end from which elements are added and/or removed is referred to as top of the stack. stacks are also referred as piles, push-down lists and last-in-first-out (lifo) lists.