DATA STRUCTURE INTERVIEW QUESTION

By-@codes.learning

Q.1. What is data structure ?

Ans. • Data structure is a fundamental concept of any programming language, essential for algorithmic design.
• It is used for the efficient organization and modification of data.

Q.2. What are the types of data structure ?

Ans. There are two types of data structure :
1. linear data structure : If the elements of a data structure result in a sequence or a linear list then it is called as linear data structure. Example : Arrays, linked list, stack, queue
2. Non linear data structure : If the elements of data structure results in a way that traver-sal of node is not done in sequential manner, then it is called non linear data structure.
Example : Trees, Graphs.

Q.3. What are applications of data structure ?

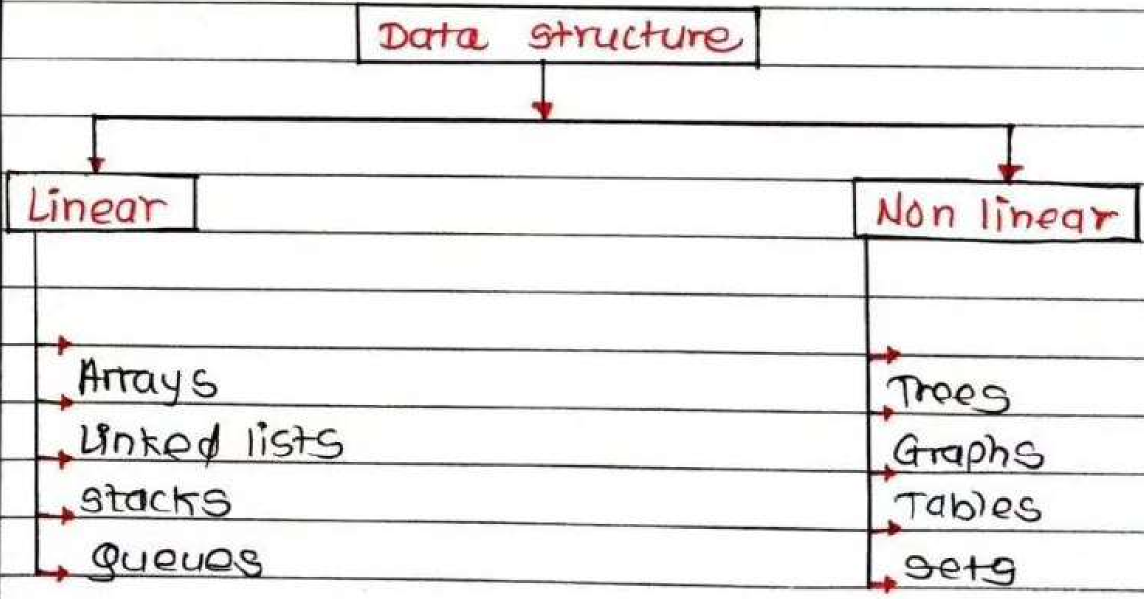
Ans. • Identifier look ups in compiler implementation are built using hash tables.
• The B-trees data structures are suitable for database implementations.

• some of the most important areas where data structures are used are as follows:

1. Artificial intelligence.
2. compiler design.
3. Machine Learning
4. Database design and management.
5. Blockchain
6. Numerical and statistical analysis.
7. Operating system development.
8. Image and speech processing
9. cryptography.

Q.4. Diagrammatic Representation of Data structures.

Ans.



(fig. Types of data structure)

Q.5. Can you explain the difference between file structure and storage structure ?

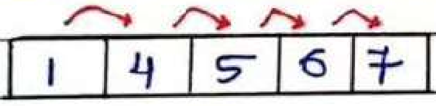
Ans. • File structure : Representation of data into secondary or auxiliary memory such as hard disk that stores data which remains intact until manually deleted is known as file structure.

- Storage structure : In this type data is stored in main memory ie. RAM. and is deleted once function that uses this data completely executed.

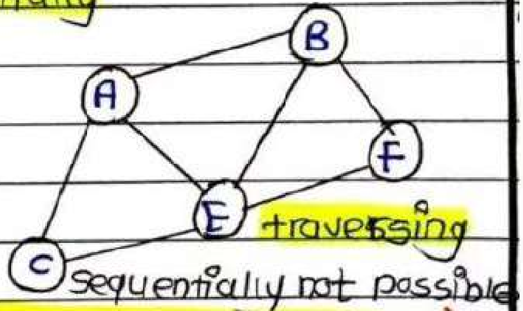
Q.6. Can you tell how linear data structure differ from non-linear data structures?

Ans.
- If elements of a data structure result in a sequence or a linear list then it is called as linear data structure, whereas traversal of nodes happens in non-linear fashion in non-linear data structures.
- Example of linear data structure : Lists, Stacks, queues, whereas graph & trees are non linear

Elements are traversed sequentially



(fig. Linear data structure)



(fig. Non linear data str.)

Q.7. What is an array?

Ans. Arrays are collection of similar types of data stored at contigous memory locations. It is simplest data structure where data element can accessed randomly just by using its index number.

Q.8. What is a multidimensional array ?

Ans. • Multi-dimensional arrays are those data structures that span accross more than one dimension.

• This indicates that there will be more than one index variable for every point of storage.

Q.9. What is a linked list ?

Ans. A linked list is a data structure that has sequence of nodes where every node is connected to next node by means of a referen-ce pointer. The elements are linked using pointer to form a chain.

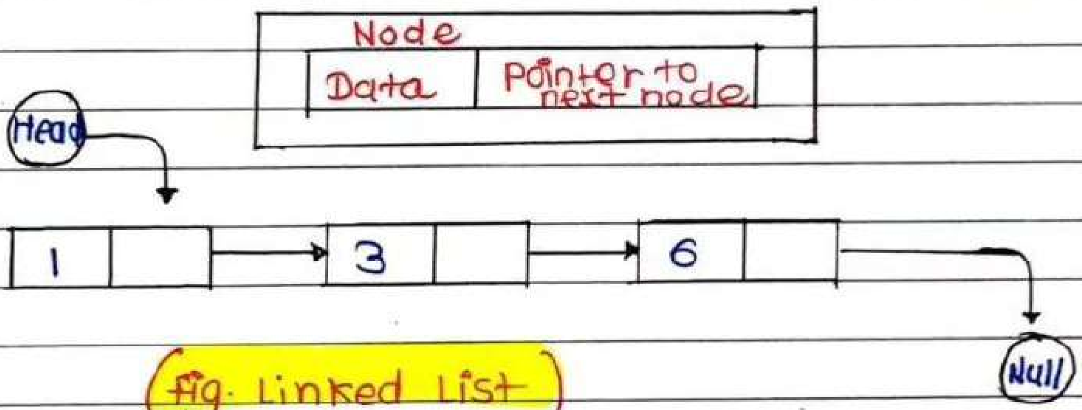


fig. Linked List

Q.10. Are linked lists of linear or non linear type ?

Ans. linked list can be considered both linear and non-linear data structure. This depends upon application that they are used for :

When linked list is used for access strategies, it is considered as linear data structure. When they are used for data storage, it can be considered as non-linear data structure.

Q.11. How are linked lists more efficient than arrays

Ans. 1. Insertion and Deletion :

Insertion and deletion process is expensive in an array as room has to be created for new elements and existing elements must be shifted.

2. Dynamic data structure :

Linked list is a dynamic data structure that means there is no need to give an initial size at time of creation.

3. No Wastage of memory :

As size of linked list can grow or shrink based on the needs of the program, there is no memory wasted coz it is allocated in runtime.

Q.12. Explain scenarios where you can use linked list and arrays.

Ans. scenarios where you can use Linked list over Array

• When we don't know exact number of elements beforehand.

• When we want to insert items anywhere in middle of list, such as implementing a priority queue, linked list is more suitable.

scenarios where we use arrays over linked list :

• When we need to index / randomly access elements more frequently.

To summarize, requirements of space, time and ease of implementation are considered while deciding which data structure has to be used over that.

@codes.learning

Q.13. What is doubly-linked list (DLL)? what are its applications?

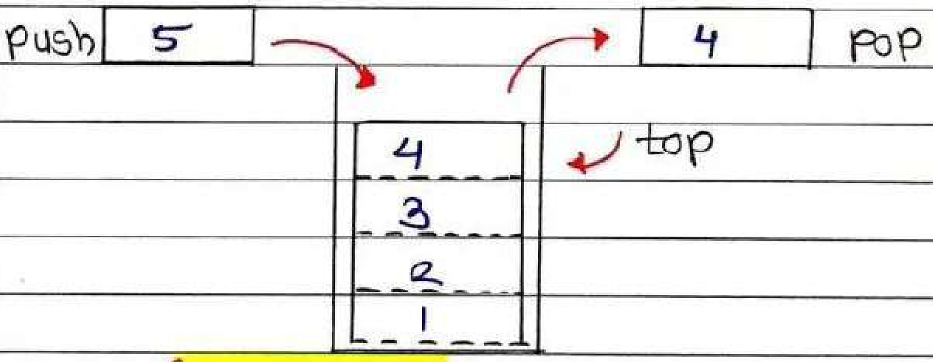
→ doubly linked list is a complex type of a linked list one that connects to next node in the sequence. Another that connects to previous. structure allows traversal of data.

Applications of DLL are:

- A music playlist with next song and previous song navigation options.
- The browser cache with BACK-FORWARD visited pages.

Q.14. What is stack? Applications?

→ Stack is a linear data structure that follows LIFO (last In first Out) approch for accessing elements. push, pop (peek) are basic operatio



(Fig. stack)

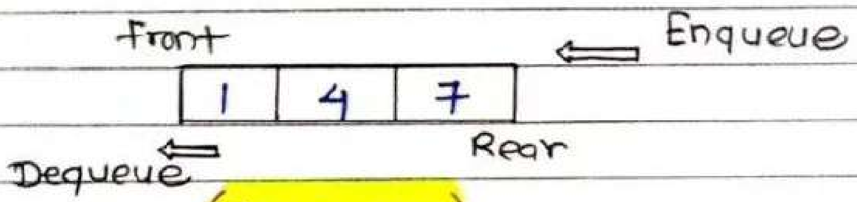
Applications of stack are as follows:

- check for balanced paranthesis in expression
- Evaluation of a postfix expression.
- problem of Infix to postfix conversion.
- Reverse a string.

@codes.learning

Q.15. What is queue ? What are applications of queue?

Ans. A queue is a linear data structure that follows FIFO (First In First Out) approach for accessing elements.

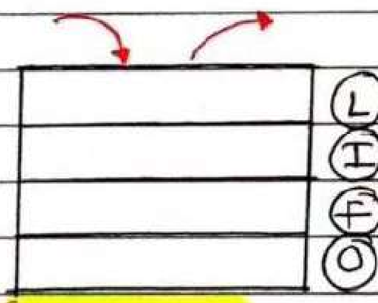


(Fig Queue)

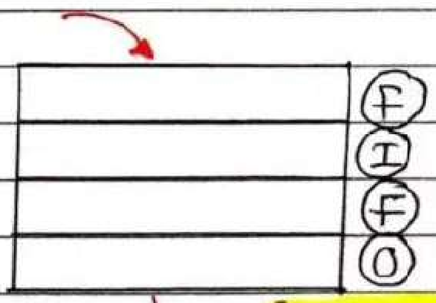
Applications of queue are as follows :
- CPU Task scheduling.
- BFS algorithm to find shortest distance
- Website request processing.

Q.16. How is stack different from a queue ?

Ans. In a stack, item that is most recently added is removed first whereas in queue, item least recently added is removed first.



(Fig. stack)



(Fig. Queue).

Q.17. Explain process behind storing a variable in memory.

Ans. A variable is stored in memory based on the amount of memory that is needed.

@codes.learning

@codes.learning

· The required amount of memory is assigned first.
· Then it is stored based on the data structu-re being used.

Q.18. What is hashmap in data structure ?
Ans. Hashmap is a data structure that uses implementation of hash table data structure which allows access of data in constant time (O(1)) complexity if you have key.

Q.19. What is requirement for an object to be used as key or value in Hashmap ?
Ans. The key | value object that gets used in hashmap must implement equals() and hashcode() method.
The hashcode is used when inserting key object into map and equals is used when trying to retrieve a value from map.

Q.20. How does Hashmap handle collisions in Java ?
Ans. The java.util.HashMap class in Java uses approach of chaining to handle collisions.
Searching will take O(n) complexity as opposed to O(1) time due to nature of linked list.

Q.21. What is time complexity of basic operations get() and put() in HashMap class ?
Ans. The time complexity is O(1) assumming that

@codes.learning

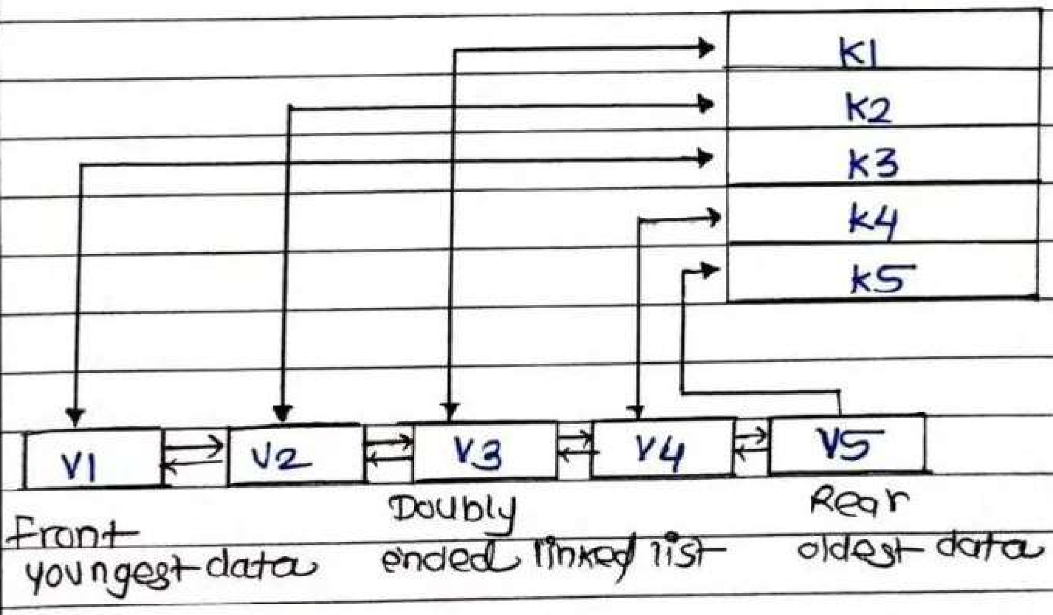
hash function used in hash map distributes elements uniformly among the buckets.

Q.22. Which data structures are used for implementing LRU cache ?

Ans. LRU cache / Least Recently Used cache allows quick identification of an element that hasn't been put to use for longest time by organizing items in order to use. Two data structures are used :

Queue : The maximum size of the queue is determined by the cache size. i.e. by total number of available frames.

HashMap : Hashmap stores page number as key along with address of corresponding queue node as value.



(fig. Hash Table)

@codes.learning

Q.23. What is priority queue?

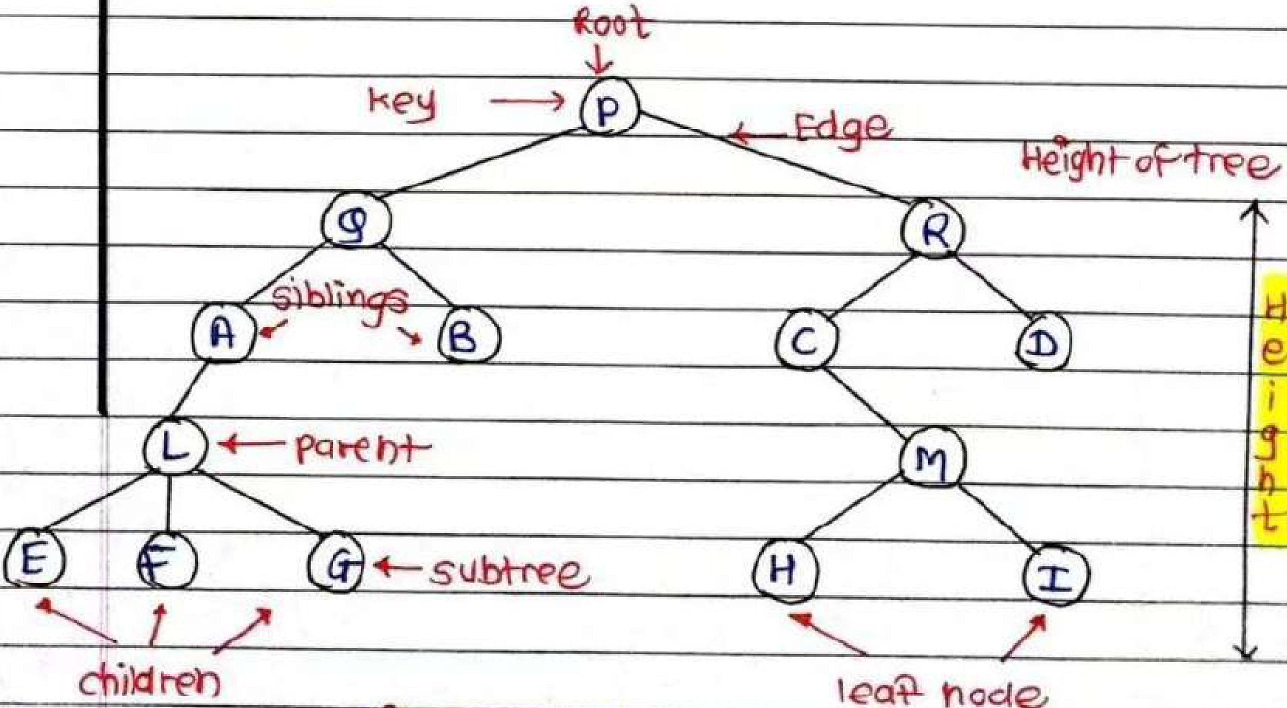
Ans. A priority queue is an abstract data type that is like a normal queue but has priority assigned to elements.
Elements with higher priority are processed before the elements with a lower priority.

Q.24. can we store a duplicate key in HashMap?

Ans. No. Duplicate keys cannot be inserted in Hashmap. If you try to insert any entry with an existing key, then old value would be overridden with new value.

Q.25. What is tree data structure?

Ans. Tree is recursive, non linear data structure consisting of set of one / more data nodes where one node is designated as root and remaining nodes are called children of root.



(fig. tree)