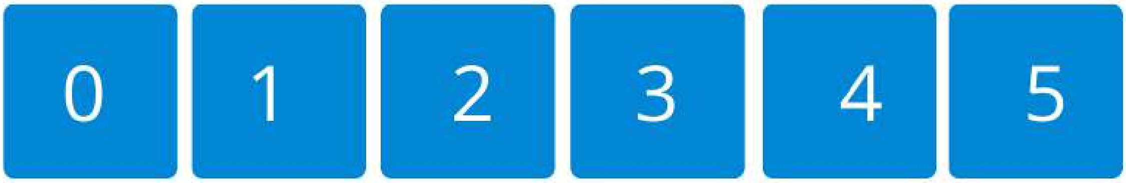
Most Asked
Data Structure
Interview QnA

@curious_.programmer

1. What is an array?

- arrays are the collection of similar types of data stored at contiguous memory locations.
- It is the simplest data structure where the data element can be accessed randomly just by using its index number

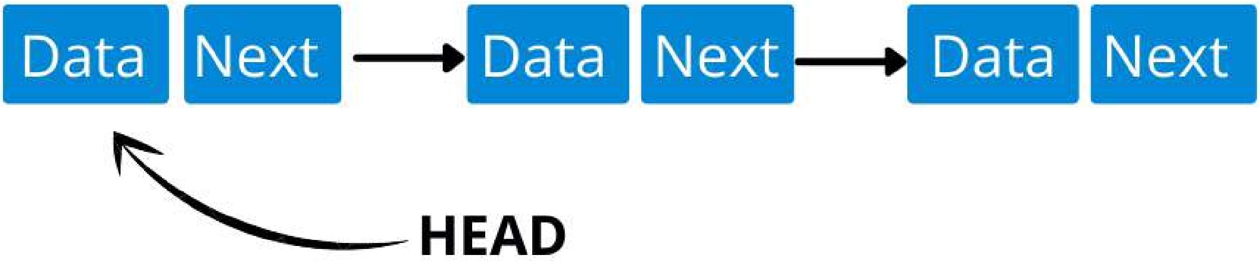


arr → array variable

[0] → Index of element to be accessed

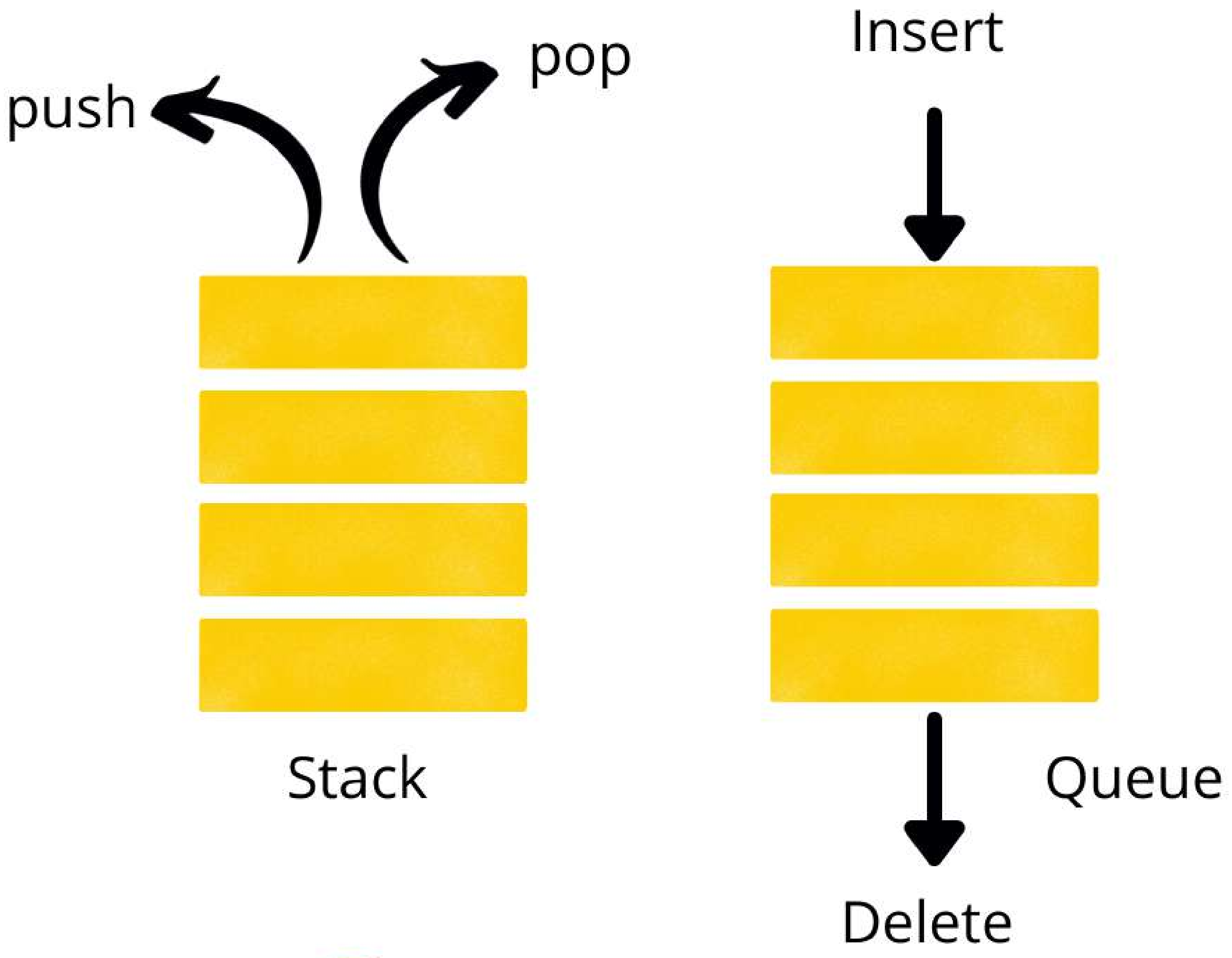
2. What is a linked list?

- A linked list is a data structure that has a sequence of nodes where every node is connected to the next node by means of a reference pointer.
- The elements are not stored in adjacent memory locations. They are linked using pointers to form a chain. This forms a chain-like link for data storage.
- Each node element has two parts:
  - a data field
  - a reference (or pointer) to the next node.



3. How is a stack different from a queue?

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



4. What is a stack? What are the applications of stack?

1. Stack is a linear data structure that follows LIFO (Last In First Out) approach for accessing elements.
2. Push, pop, and top (or peek) are the basic operations of a stack

applications of a stack:

- Check for balanced parentheses in an expression
- Evaluation of a postfix expression
- Problem of Infix to postfix conversion
- Reverse a string

5. What is a multidimensional array?

- Multi-dimensional arrays are those data structures that span across more than one dimension.
- This indicates that there will be more than one index variable for every point of storage. This type of data structure is primarily used in cases where data cannot be represented or stored using only one dimension. Most commonly used multidimensional arrays are 2D arrays.
- 2D arrays emulates the tabular form structure which provides ease of holding the bulk of data that are accessed using row and column pointers.

6. What is a queue? What are the applications of queue?

1. A queue is a linear data structure that follows the FIFO (First In First Out) approach for accessing elements.
2. Dequeue from the queue, enqueue element to the queue, get front element of queue, and get rear element of queue are basic operations that can be performed.

applications of queue are:

- CPU Task scheduling
- BFS algorithm to find shortest distance between two nodes in a graph.
- Website request processing
- Used as buffers in applications like MP3 media player, CD player, etc.
- Managing an Input stream

7. Can you explain the difference between file structure and storage structure

- File Structure: Representation of data into secondary or auxiliary memory say any device such as a hard disk or pen drive that stores data that remains intact until manually deleted is known as a file structure representation.
- Storage Structure: In this type, data is stored in the main memory i.e RAM, and is deleted once the function that uses this data gets completely executed.
- The difference is that the storage structure has data stored in the memory of the computer system, whereas the file structure has the data stored in the auxiliary memory.

8. How are linked lists more efficient than arrays?

- **Insertion and Deletion:** The insertion and deletion process is expensive in an array as the room has to be created for the new elements and existing elements must be shifted.
- **Dynamic Data Structure**: The linked list is a dynamic data structure which means there is no need to give an initial size at the time of creation as it can grow and shrink at runtime by allocating and deallocating memory. Whereas, the size of an array is limited as the number of items is statically stored in the main memory.
- **No wastage of memory**: As the size of a linked list can grow or shrink based on the needs of the program, there is no memory wasted because it is allocated in runtime.

9. Explain the process behind storing a variable in memory.

Declaration: Before a variable can be stored in memory, it must first be declared. This involves specifying the data type of the variable and giving it a name.

Memory allocation: Once the variable is declared, the program needs to allocate memory to store it. This is typically done automatically by the compiler or interpreter. The amount of memory allocated depends on the data type of the variable. For example, an integer typically requires 4 bytes of memory.

Assignment: After memory has been allocated for the variable, you can assign a value to it. This is done using the assignment operator "=".