HyperionDev

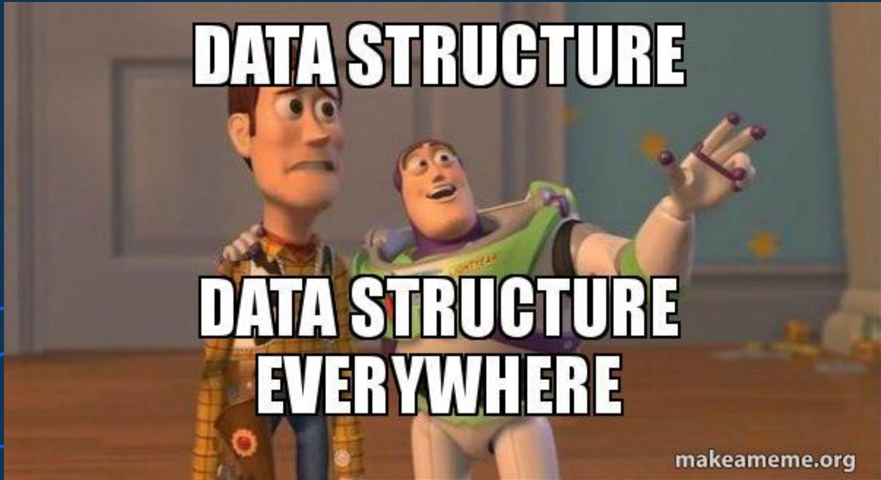
Linear Data Structures

Today's concepts

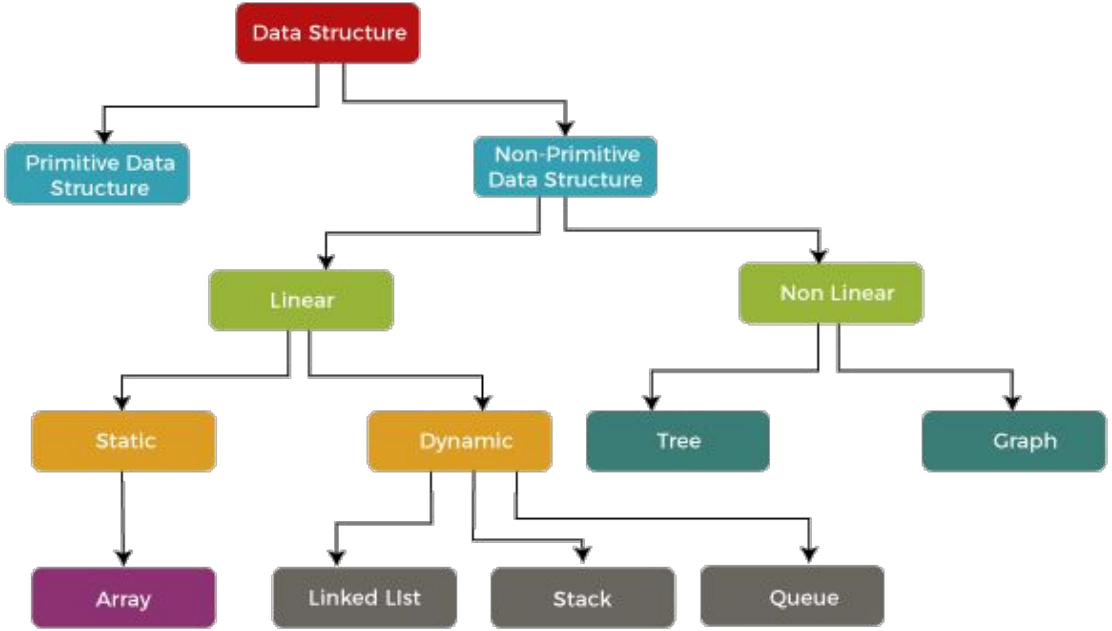
- Arrays
- Linked Lists
- Arrays vs Linked Lists

What is a data structure?

A **data structure** is a particular way of organizing data in a computer so that it can be used effectively.

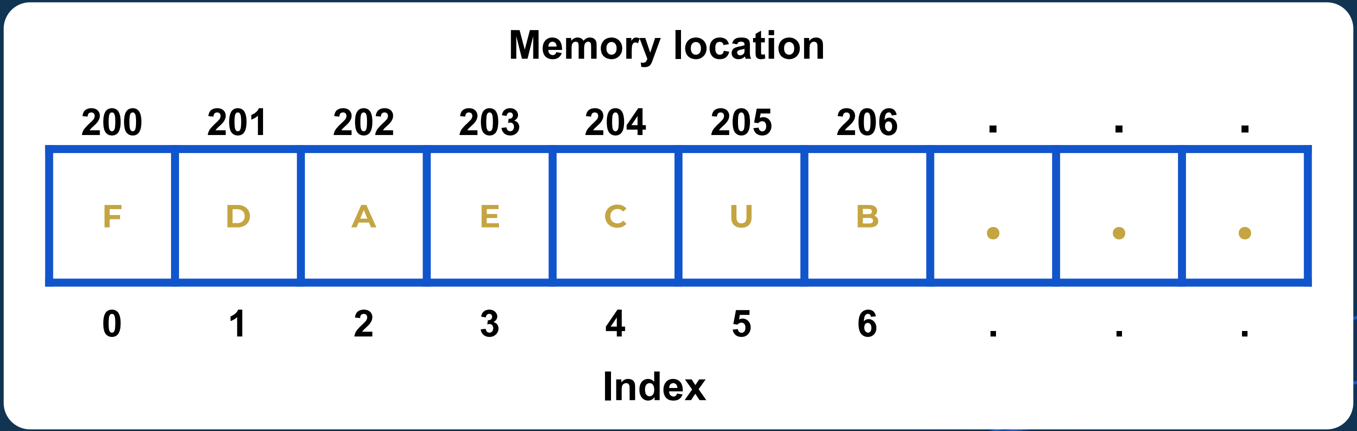


Types of data structures



Arrays

An **array** is a collection of items stored at contiguous memory locations.



Memory location

200	201	202	203	204	205	206	.	.	.
F	D	A	E	C	U	B	.	.	.
0	1	2	3	4	5	6	.	.	.

Index

Static Arrays

A **static array** is an array for which the size or length is determined when the array is created and/or allocated.

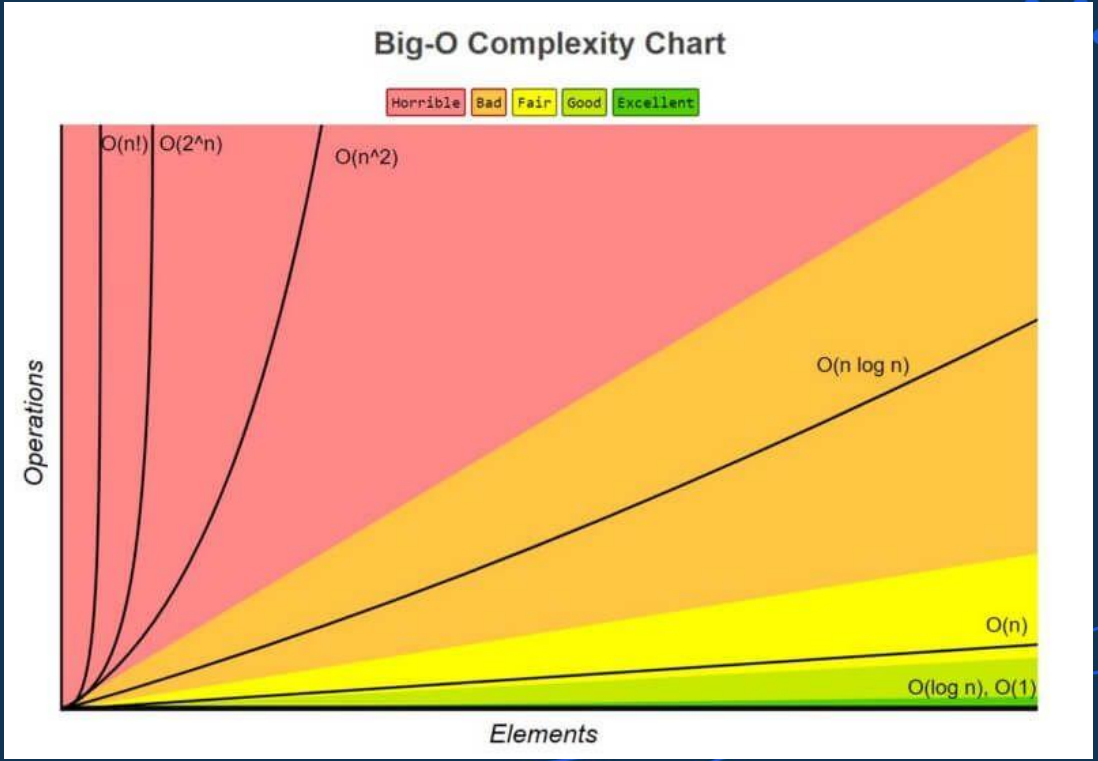
Language	**Defined values**	**Fixed with undefined values**
C++	int values[] = {0, 1, 2};	int values[3];
Java	int values[] = {0, 1, 2};	int[] values = new int[3];
Javascript	let values = [0, 1, 2]	let values = new Array(3)
Python	values = [0, 1, 2]	values = [None] * 3

Dynamic Arrays

Dynamic arrays allow elements to be added and removed at runtime. Most current programming languages include built-in or standard library functions for creating and managing dynamic arrays.

Language	**Class**	**Add**	**Remove**
C++	#include <list> std::list	insert	erase
Java	java.util.ArrayList	add	remove
Javascript	Array	push	pop, slice, splice
Python	List	append	remove

Big O Notation



Time complexities of array operations

	Array
Accessing elements	O(1)
Insert / remove from beginning	O(n)
Insert / remove from end	O(1)
Insert / remove from middle	o(n)

[10, 9, 6, 5, 8]

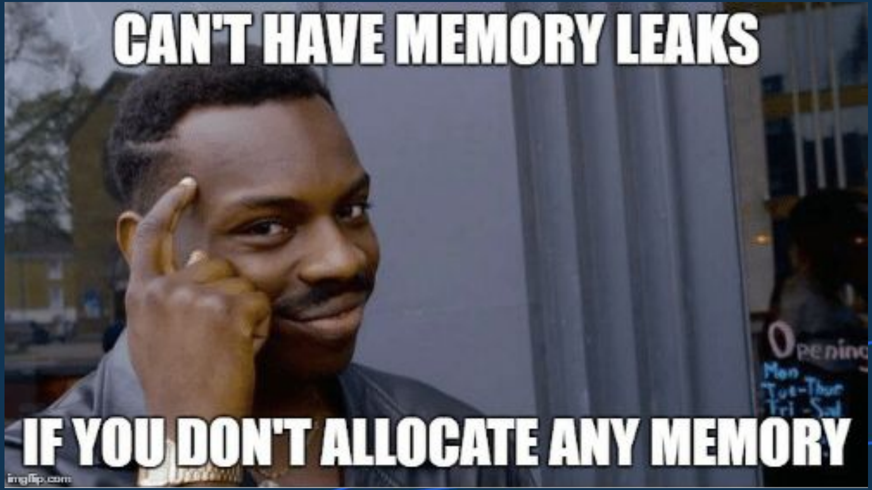
Advantages of Arrays

- Code optimization
- Functionality
- Index-Based
- Multi-dimensional
- Memory allocation
- Multiple uses

Does your dog bite?
No, but it can hurt you in other ways.
Array indexing starts at 1

Disadvantages of Arrays

- Size is fixed
- The problem in expansion
- Memory wastage
- Limitation of type of data
- Operational limitation
- Memory space
- Index bound checking



Real world use cases of Arrays

- Arrangement of leader-board of a game
- Image processing
- Speech processing
- Contact applications
- Songs playlists
- Used to implement strings

Quiz: Arrays

Elements in an array are accessed _____:

👍Sequentially

👎Randomly

Quiz: Arrays

Elements in an array are accessed _____:

👍Sequentially

👎Randomly ✅

Quiz: Arrays

In general, the index of the first element in an array is _____:

👍1

👎0

Quiz: Arrays

In general, the index of the first element in an array is _____:

👍1

👎0 ✅

Quiz: Arrays

What are the advantages of arrays?

👍Easier to store elements of same data type

👎Elements in an array cannot be sorted

Quiz: Arrays

What are the advantages of arrays?

👍Easier to store elements of same data type ✅

👎Elements in an array cannot be sorted

Quiz: Arrays

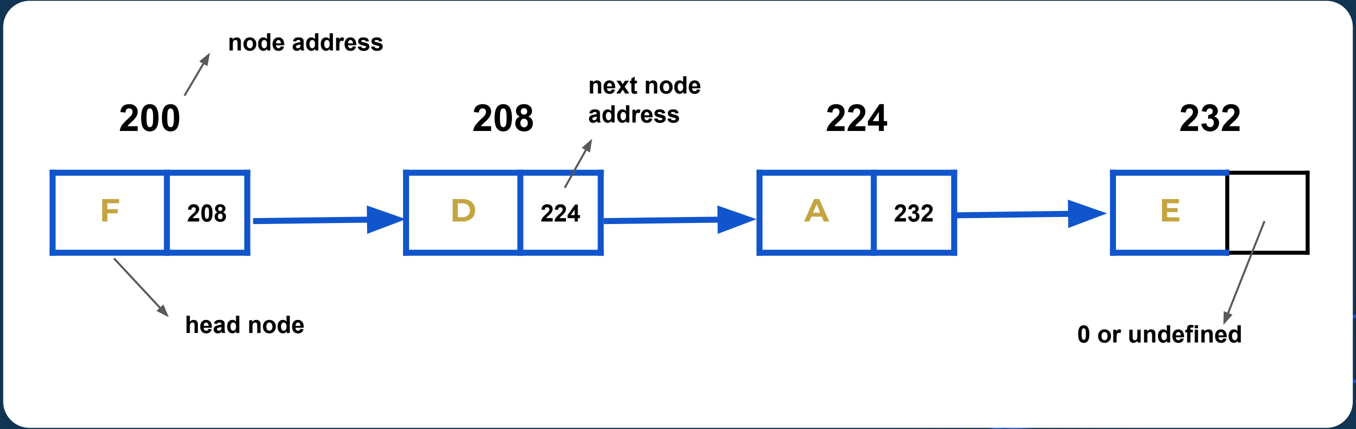
Which of these best describes an array?

👍 Container of objects of similar types

👎Arrays are immutable once initialised

Linked lists

A **linked list** is a linear data structure, in which the elements are not stored at contiguous memory locations.



Linked list implementation

JavaScript

```
class LinkedList {
  constructor(head = null) {
    this.head = head
  }
}

class Node {
  constructor(value, next = null) {
    this.value = value
    this.next = next
  }
}
```

Python

```
class Node:
  def __init__(self, data):
    self.data = data
    self.next = None

class LinkedList:
  def __init__(self):
    self.head = None
```

Operations on Linked List

Examples of basic operations:

- **Insertion -** add an element at the beginning of the list.
- **Deletion -** deletes an element at the beginning of the list.
- **Display -** show the complete list.
- **Search -** search an element using a given key.
- **Delete -** delete an element using a given key.

Linked list: Insertion

JavaScript

```
class Node {
  constructor(value, next = null) {
    this.value = value
    this.next = next
  }
}

class LinkedList {
  constructor(head = null) {
    this.head = head
  }

  function push(new_data) {
    let new_node = new Node(new_data);
    new_node.next = head;
    head = new_node;
  }
}
```

Python

```
class Node:
  def __init__(self, data):
    self.data = data
    self.next = None

class LinkedList:
  def __init__(self):
    self.head = None

  def push(self, new_data):
    new_node = Node(new_data)
    new_node.next = self.head
    self.head = new_node
```

Linked list: Deletion

JavaScript

```javascript
function deleteNode(key) {
    let temp = head, prev = null;

    if (temp != null && temp.data == key) {
        head = temp.next;
        return;
    }

    while (temp != null && temp.data != key) {
        prev = temp;
        temp = temp.next;
    }

    if (temp == null)
        return;

    prev.next = temp.next;
  }
```

Python

```python
def deleteNode(self, key):
    temp = self.head
    if (temp is not None):
        if (temp.data == key):
            self.head = temp.next
            temp = None
            return

    while(temp is not None):
        if temp.data == key:
            break
        prev = temp
        temp = temp.next

    if(temp == None):
        return

    prev.next = temp.next
    temp = None
```

Linked list: Search

JavaScript

```javascript
function search( head , x)  {
    let current = head;
    while (current != null) {
        if (current.data == x)
            return true;
        current = current.next;
    }
    return false;
}
```

Python

```python
def search(self, x):
        current = self.head

        while current != None:
            if current.data == x:
                return True

            current = current.next

        return False
```

Time complexities of linked lists operations

	Array
Accessing elements	O(n)
Insert / remove from beginning	O(1)
Insert / remove from end	O(n)
Insert / remove from middle	o(n)

Real world use cases of Linked lists

- Image viewer software
- Web pages
- Music players
- UNDO, REDO or DELETE operations.

Arrays vs Linked lists

- Memory allocation
- Memory efficiency
- Execution time
- Insertion
- Dependency
- Size

Quiz: Linked List

A linked list is a linear data structure, in which the elements are stored at contiguous memory locations.

👍True

👎False

Quiz: Linked List

A linked list is a linear data structure, in which the elements are stored at contiguous memory locations.

👍True

👎False ✅

Quiz: Linked List

A node contains two fields i.e. data stored at that particular address and the pointer which contains the address of the next node in the memory.

👍True

👎False

Quiz: Linked List

A node contains two fields i.e. data stored at that particular address and the pointer which contains the address of the next node in the memory.

👍True ✅

👎False

Quiz: Linked List

Which of these is not an application of a linked list?

👍Random Access of elements

👎 To implement file systems

Quiz: Linked List

Which of these is not an application of a linked list?

👍Random Access of elements ✅

👎 To implement file systems

Challenge:

Arrays

- Find the minimum and maximum element in an array without array built-in methods
- Write a program to reverse the array without built-in array methods
- Write a program to sort the given array without built-in array methods

Linked lists

- Write a function to reverse the nodes of a linked list.
- Delete middle of linked list
- Write a function to check if a linked list is a palindrome

THANK YOU

HyperionDev