code decoded

data structures

linked list

A LinkedList is an ordered collection of data where the memory is allocated dynamically.

A LinkedList is composed of Nodes. A Node is a structure that contains

a) data, content of some kind
b) next, a reference to another Node

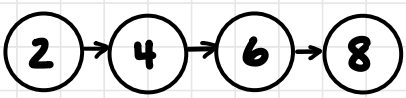
Instead of being stored contiguously, each Node contains the address of the following Node.

```
class Node {
    int data;
    Node *next;
}
```

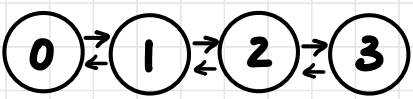
```
class LinkedList {
    Node *head;
}
```

A LinkedList is a reference to the first Node.

In a Singly Linked List, each Node contains one reference – the address of the next Node.



In a Doubly Linked List, each Node contains two references – the address of the previous Node & the next Node.



SEARCH O(N)

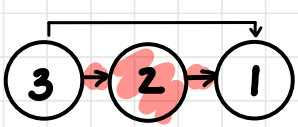
Accessing a node can only be done by traversing to the node.

```
Node current = root;
while (current.value != searchValue) {
    current = current.next;
}
```

DELETE AT K O(N)

To delete a Node, we first have to locate it by traversing to K.

Save K.next to a variable.
Set K-1.next to that variable.



INSERT AT K O(N)

To insert a node at a specific position K, we must first traverse to K.
Create a new Node instance.
Set the new node's next to node K-1's next.
Set node K-1's next to point to the new node.