Linked Lists

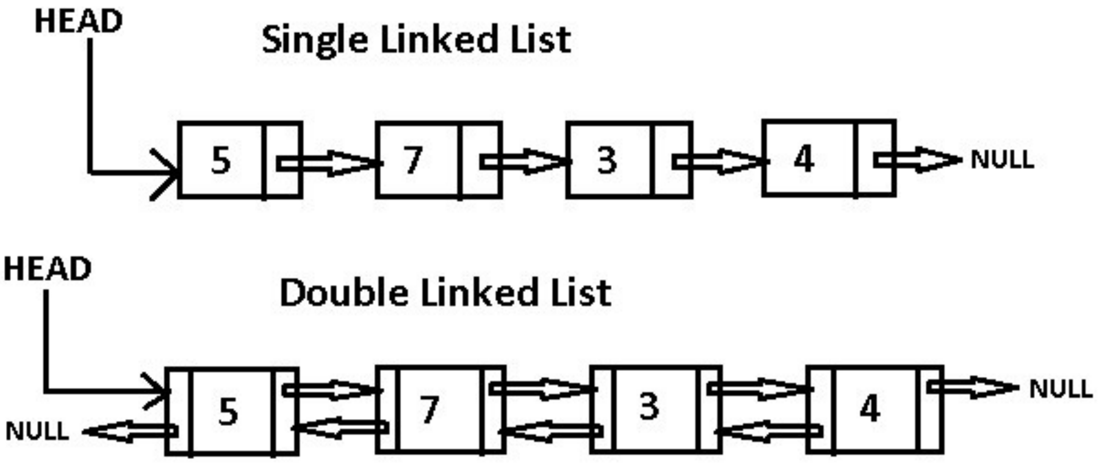
They're like lists, but linked

	Insertion	Deletion	Search	Space
Complexity:	O (1)*	O (1)*	O (n)	O (n)

* Provided you have a reference to the node/location you want to be deleted/inserted. Otherwise, you will have to traverse the list, which is O (n)

Description:

A linked list is simply a sequence of nodes. Each node consists of a value and one or two pointers.

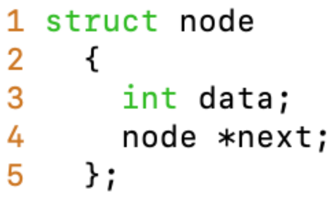


Source: https://medium.com/journey-of-one-thousand-apps/data-structures-in-the-real-world-508f5968545a

Implementation:

A linked list can be implemented easily with a single node class*. Your node object should have a data attribute and a next attribute that is a node. You can add more methods for insertion and deletion depending on your use case, but they are not necessary.

Ex.



```
struct node
  {
    int data;
    node *next;
  };
```

Applications:

Storing utterable data that gets appended often and other future data structures. Ex. Image viewers, storing forward/backward pages in a browser, stacks, queues, hash maps

Know how to:

- Mutate a linked list (Append, delete, change value)
- Loop/ recurse through a linked list

Details:

- Head: The head is the first item of a linked list, it is used to keep track of the start of a linked list
  - * If you are just using a node class, then trouble may arise if you have references to a head and then that head changes. This can be mitigated by creating a wrapper class that keeps track of the head.

Arrays vs Linked Lists

- Similarly to arrays, access to a linked list is O(1). However unlike arrays, you may only access the current nodes data and must iterate to other values.
- You can append a linked list in O(1) time, unlike arrays which can not be appended.

Tips

- When iterating/recursing through a linked list, the end/base case is when the current node or current node's next is equal to Null
- Sometimes it may be helpful to have two pointers in an array moving at different speeds

Interview Problems:

- Return the Kth to last element of a singly linked list
- Given two linked lists, return the smallest node.