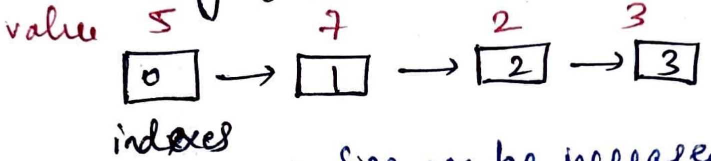
Intro to linked list

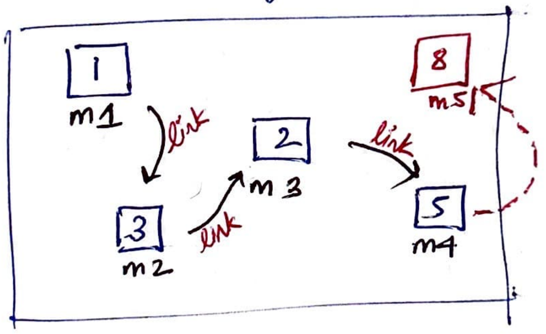
★ What is a linked list ?

★ Why not arrays ?

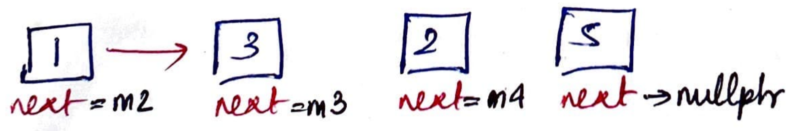
- Array is of fixed size and no extra element can be added if needed i.e size of array cannot be increased/decreased.
- Arrays are stored in a contigious locations. This means that arrays are stored in a single block of memory. In this block, elements are stored one after the other without any gaps between them.



Linked list : → size can be increased at any location.
→ Not in contiguous locations
↓
Can store any datastructure like arrays



m1 → memory location 1



head = m1 , tail = m4

★ Struct/Class in C++ / Java ★

int x = 2 ; → creates x in the heap (m1) memory & stores value 2 in it

int* y = &x ; → address of x

cout<< y ;

& in C++, address of x cannot be stored in a variable, thus we store a pointer to the memory location

int * y → integer type pointer 'y'.

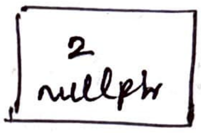
Struct → is a self-defined datatype.

```
Struct Node {
    int data;
    Node* next;
    Node (data1, next1) {
        data = data1;
        next = next1;
    }
};
```
✓

Note: Can use Class in place of Struct to use OOP based concepts

data
next
Node

```
Node x = Node (2, nullptr);
Node* y = &x } pointer y pointing to memory location of x
```



```
Node* y = new Node (2, nullptr);
```
✓

```
cout << y → data
```

automatically stores a pointer to the memory location of y.

★ **MEMORY SPACE USED :-**

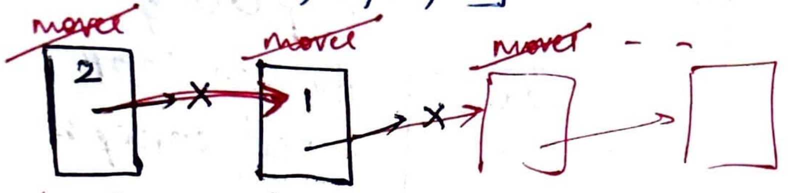
↳ depends on the systems

32-bit	64-bit
int → 4 bytes	int → 4 bytes
* → 4 bytes	* → 8 bytes
8 bytes	12 bytes

(not imp)

★ How to convert Array to LL?

arr [] = [2, 1, 3, 8]



head
↓
is the starting point.

temp
new node (1, null)

mover → next = temp

★ NEVER TAMPER THE HEAD
↓
eelse you would miss the starting location

```
Node* convertArr2LL(vector<int> &arr) {
        Node* head = new Node(arr[0]);
        Node* mover = head;
        for (int i=1; i<arr.size(); i++) {
            Node* temp = new Node(arr[i]);
            mover → next = temp;
            mover = mover → next;
        }
        return head;
}
```

Leetcode Syntax

```
class Node {
    public:
    ~~Node~~ T data;
    Node <T> * next;
    ;
    ;
    ;
```

→ data can be of any variable - int, string etc.

specify like this.

```
int searchLL (Node<int> * head, int k) {
    Node<int>* temp = head;
    :
    :
```