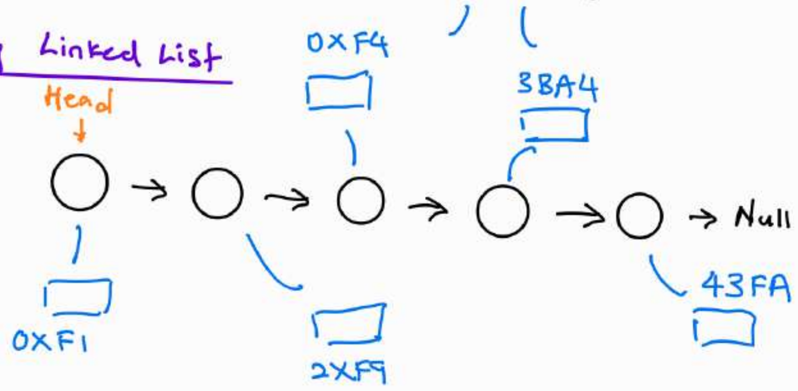
Linked Lists

head, tail, next, cur, prev

- made of Nodes & pointers.
- node store the memory address of data. (not the data itself)
- Not stored in contiguous memory

- can expand Linkedlist by dynamically allocating memory

Singly Linked List



- Node only points to next node (no prev node ref.) i.e only can move in 1 dirn

Linked List Complexity

Insertion :

- Random insertion : O(n) : move cur pointer to desired location
- Insert front : O(1) : No Moving of element required.
- insert end : O(n) for singly , O(1) with tail ptr
  - array was O(1) for end , O(n) for front

Delete :

- Delete front : O(1) : No shifting element required
- Delete end : O(n) for singly , O(1) if ∃ tail ptr

Search :

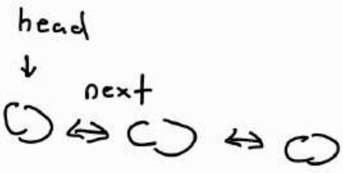
- O(n) for both sorted & unsorted

Access :

- O(n) , array was O(1)

Benefit of Linked List : Dynamically assigned memory

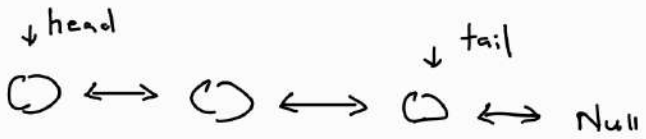
Doubly Linked List



- next , prev ptr
- can traverse both dirns.
- does not speed up algo, just make coding easier.

Tail Pointer

- Point to end of Linked List



- makes insert & delete at end = O(1) instead of O(n)