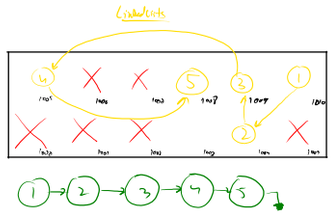
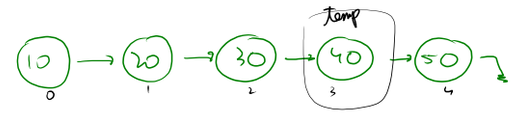
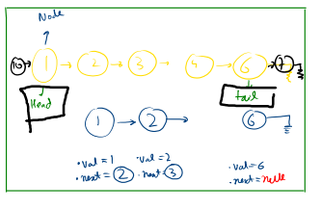
Linked Lists





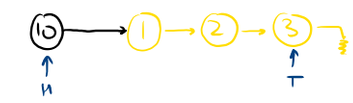

```
let temp = head
let i = 0
while ( i < idx){
    i++
    temp = temp.next
}
return temp
```

getAt(3) => 40



- getSize ✓
- addFirst ✓
- addLast ✓
- display ✓
- removeFirst ✓
- removeLast ✓
- getAt(idx)
- addAt(idx)
- removeAt(idx)
- reverse

10 20



(val) = 10

- New node
- NN.next = head
- head = NN
- size++

if(size = 0)

Head = NN
Tail = NN

H 10 T Size = 1

```
while( temp.next != tail){
    temp = temp.next
}
temp.next = null
this.tail = temp
this.size --
```

- Create node
- T.next = nn
- T = nn
- size++