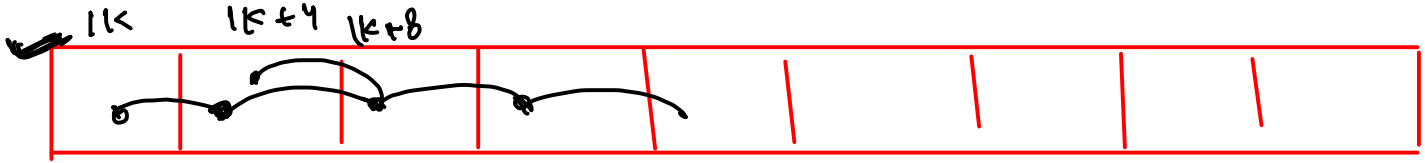
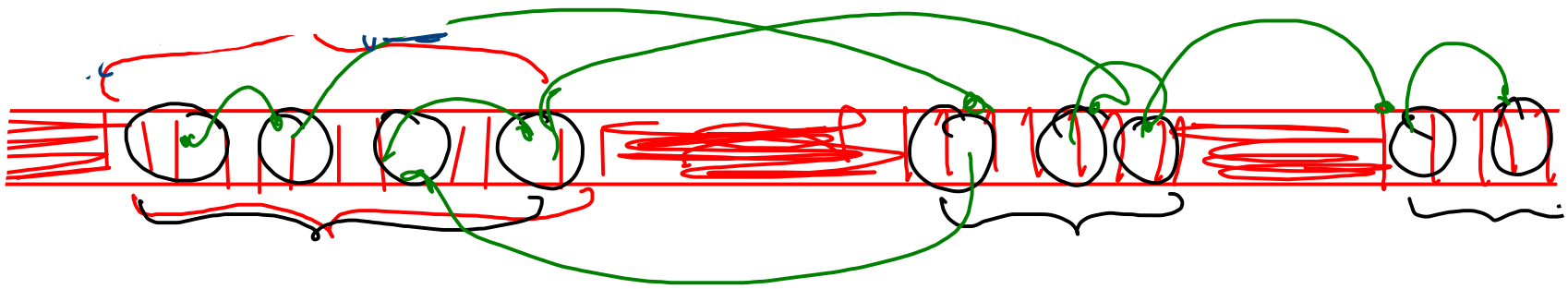
Linked list

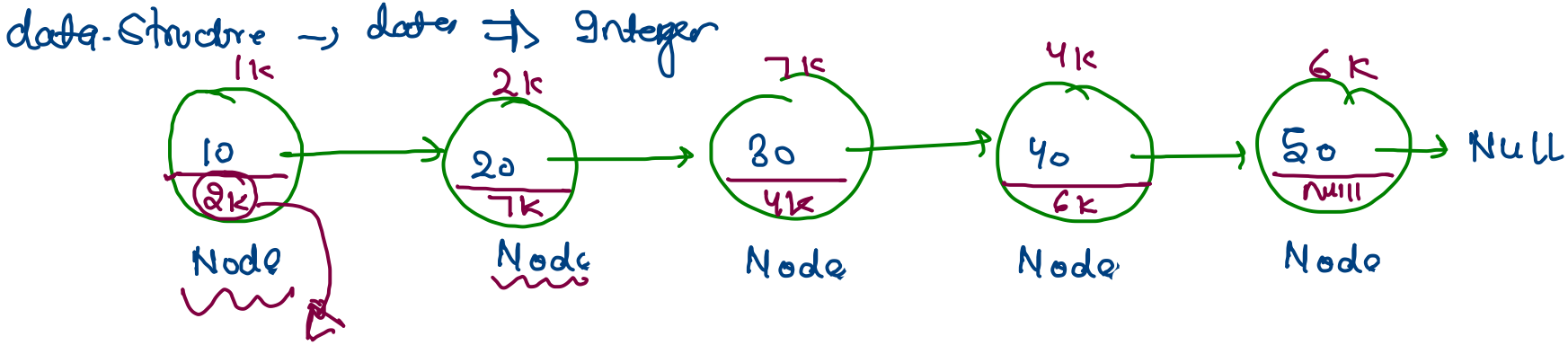
Data Structure — Time and Space

1. Array + Array List
2. Stack
3. Queue



Continuous Memory.



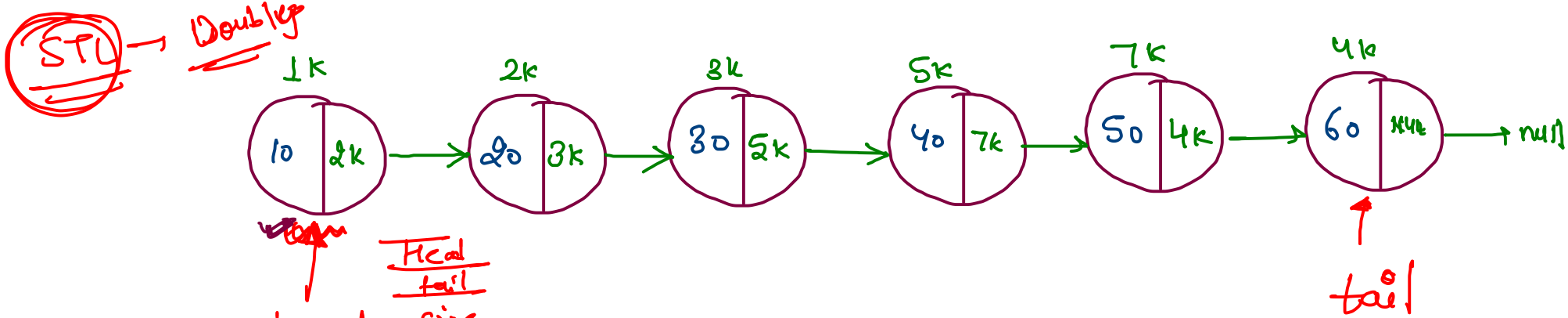


Reference (Address)
↓
type of address
↓
Node

Node → { int data
Node next

↑
user defined data type

Student S7 = new Stud



Starting of Linked List

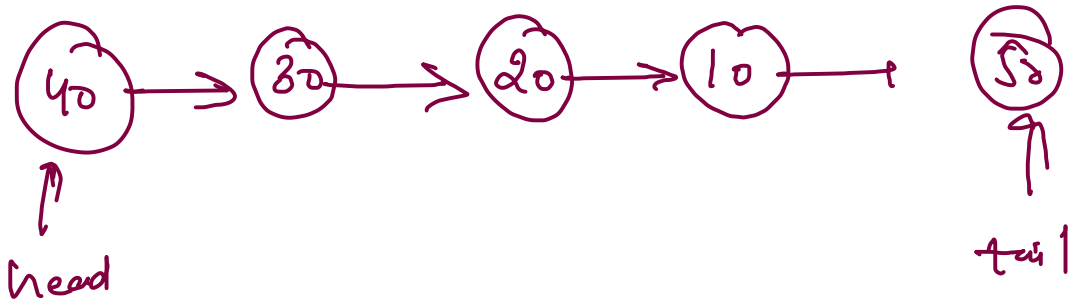
Linked List

- addFirst
- addLast
- addAt
- removeFirst
- removeLast
- removeAt
- getFirst
- getLast
- getAt

- size
- display

capacity constraint

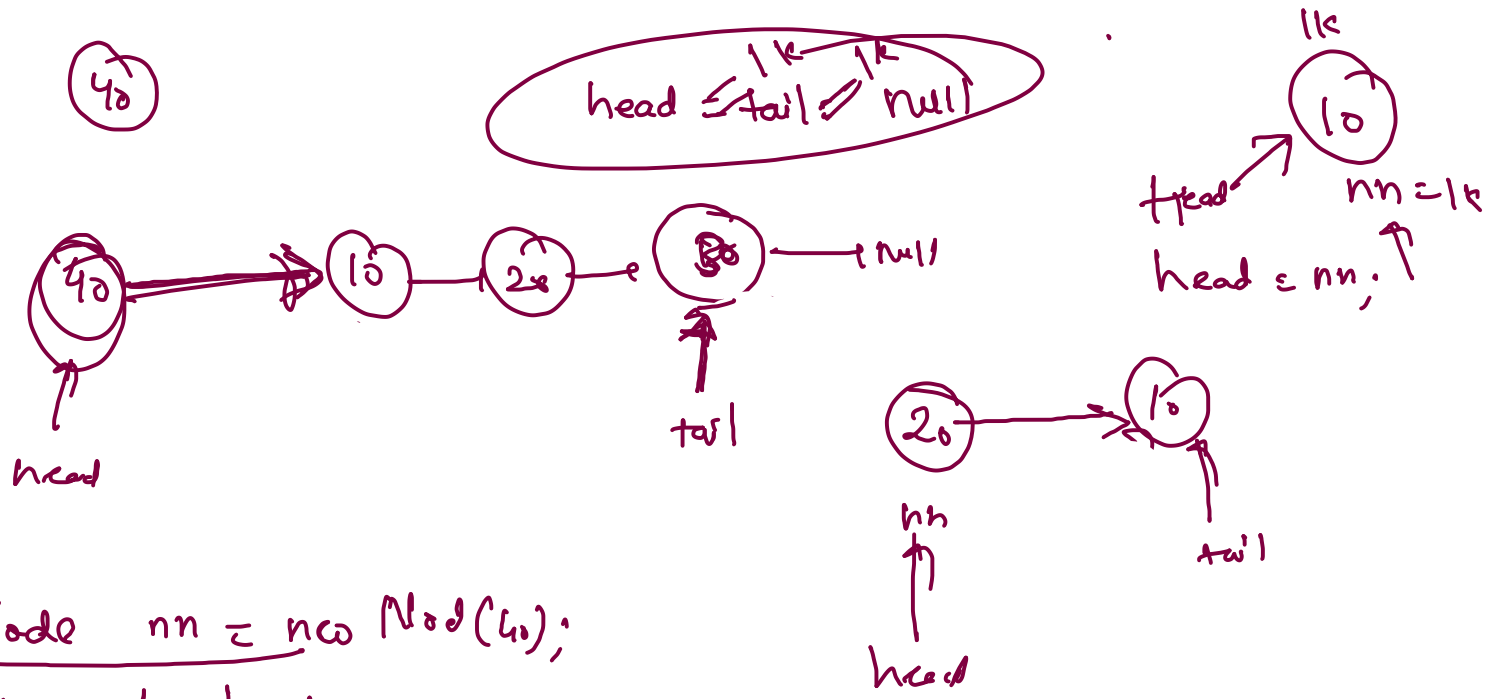
Memory available




```
list.addFirst(10)
list.addFirst(20)
list.addFirst(30);
list.addFirst(40);
list.addLast(50)
list.display()
```

40 → 30 → 20 → 10 → 50 → null

addFirst



Node — Node nn = new Node(40);

Connect — nn.next = head;

Assigning — head = nn.

Size — Size ++;

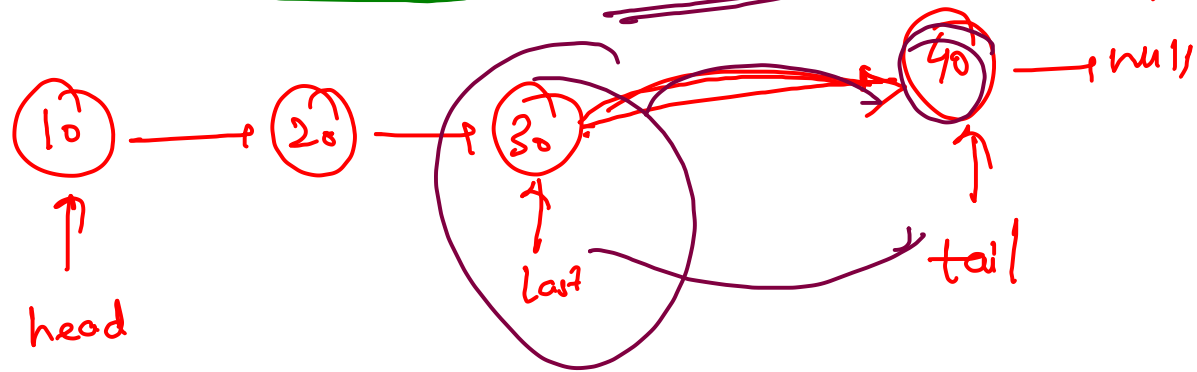
addLast

addLast(10);

tail

addLast( ) → O(1)

O(n) ← tail → getLastNode
addLast( )



Node nn = new Node(10);

tail.next = nn;

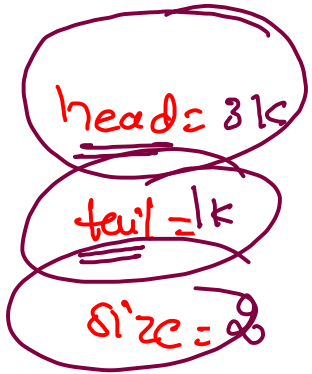
tail = nn;

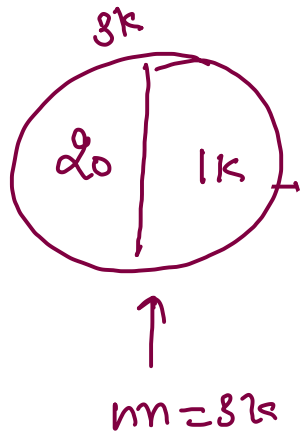
size++;

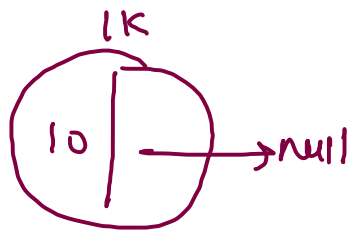
head = null

tail = null

head.next = 

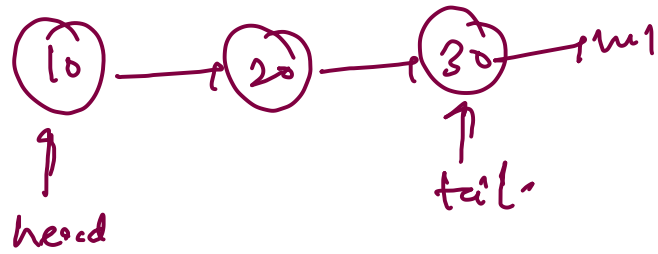


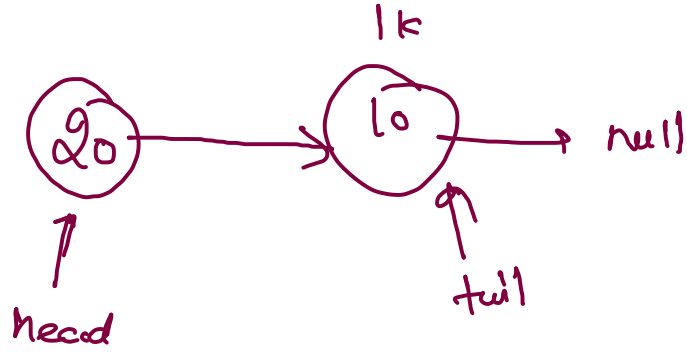




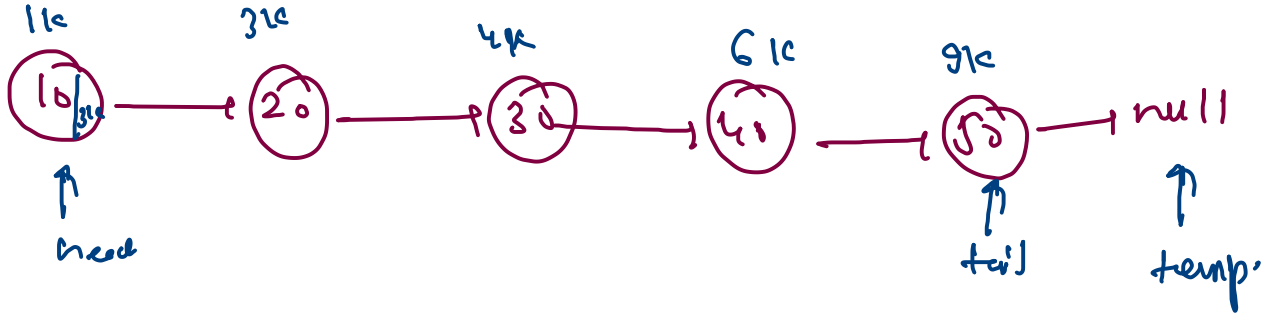

```
Node nn = new Node(val);
nn.next = this.head;
this.head = nn;
this.size++;
```

addFirst(10)
addFirst(20)





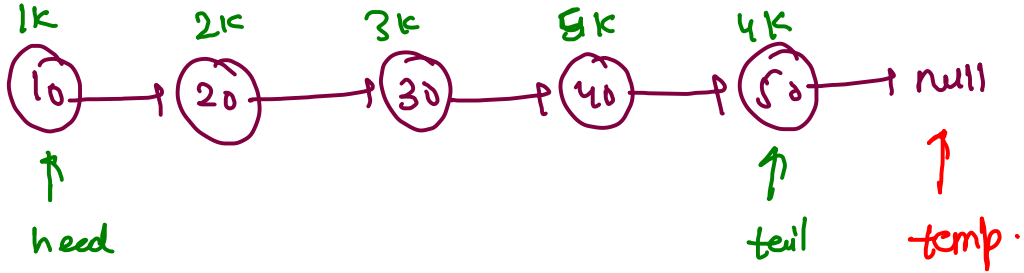
Display:




```
Node temp = head;

while( temp != null) {
    Syso( temp.data + "->");
    temp = temp.next;
}
Syso ("null");
```

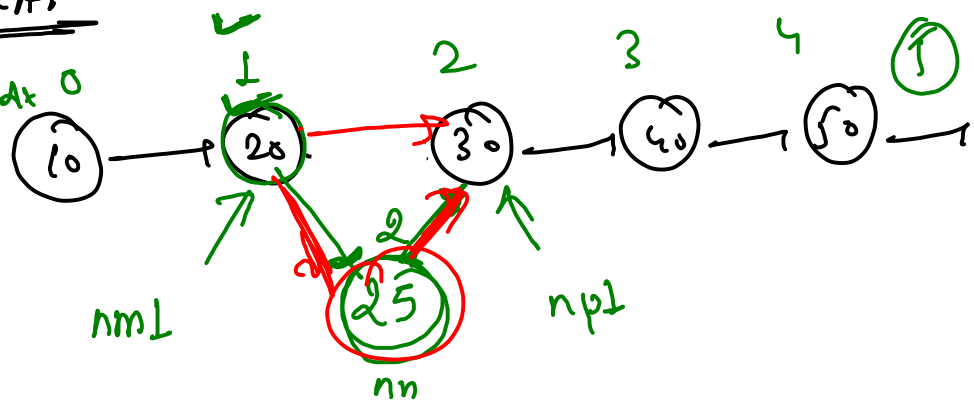
10 → 20 → 30 → 40 → 50 → null




```
public void display() {
    Node temp = this.head;
    while(temp != null) {
        System.out.print(temp.data + " -> ");
        temp = temp.next;
    }
    System.out.println("null");
}
```

10 → 20 → 30 → 40 → 50 → null

AddAt-



addAt(25,2)

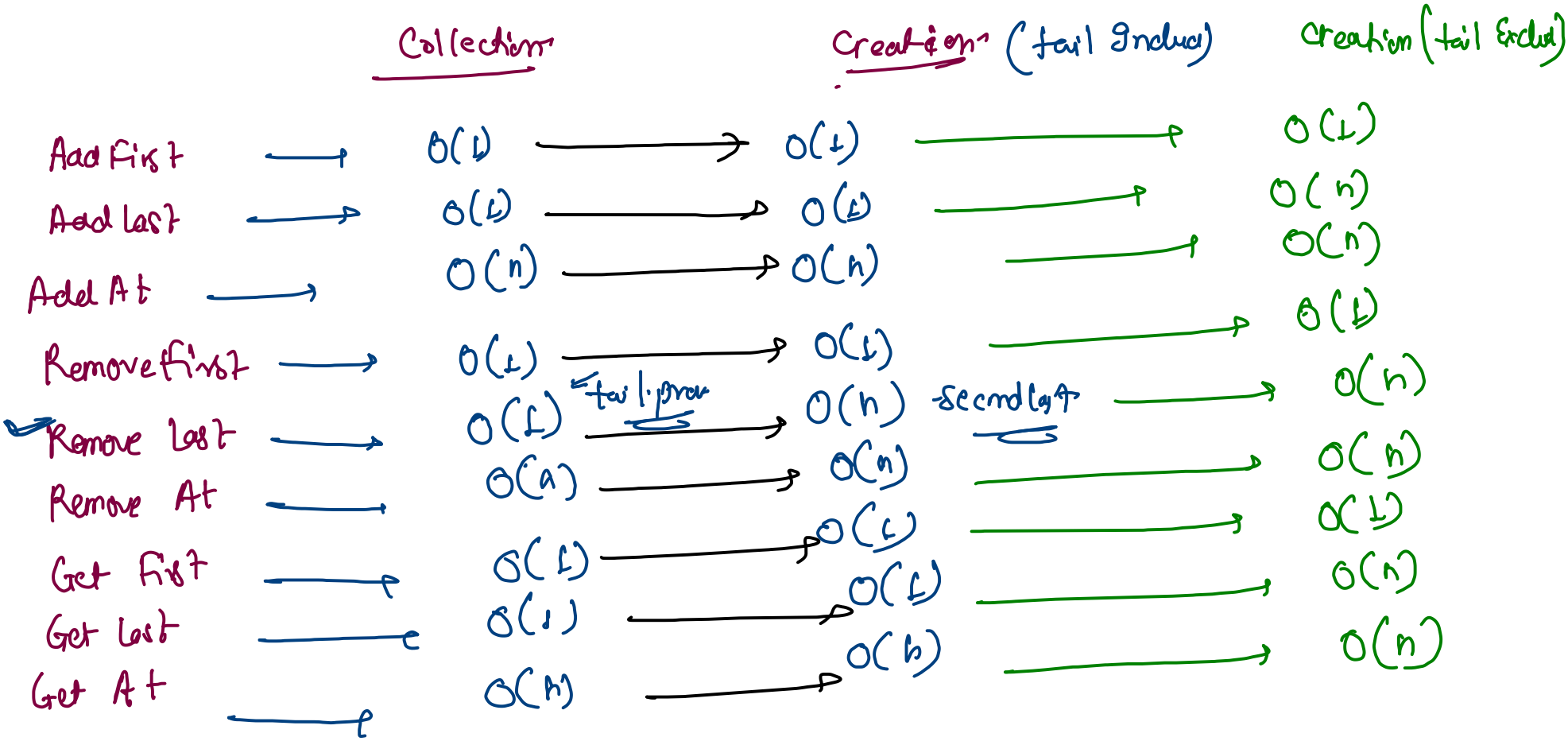
Node nm1 = GetNode( );

nn.next = nm1.next

nm1.next = nn;

Imax = 0

Imax = 5


Collection
Creation (tail Included)
Creation (tail Excluded)
Add First
Add last
Add At
Remove first
Remove last
Remove At
Get first
Get last
Get At
O(1)
O(1)
O(n)
O(1)
O(1)
O(n)
O(n)
O(1)
O(n)
O(1)
O(1)
O(n)
O(1)
O(n)
O(n)
O(1)
O(n)
O(n)
O(n)
O(n)
O(n)
O(n)
O(1)
O(n)
O(n)
tail.prev
second last