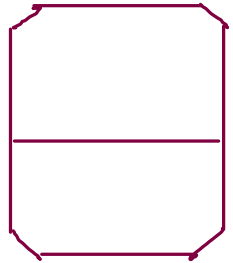
LRU Cache

Recent tabs → limited Recent Apps

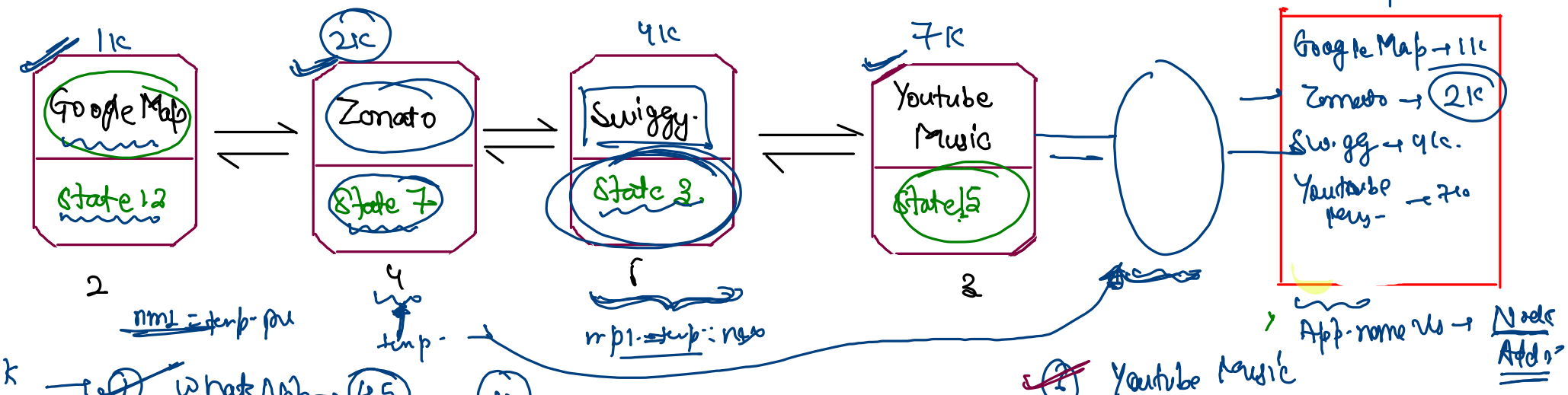
Mobile

(1) More than Limit.
→ Remove Head
temp = map.get(Zomato)

limit = 4.] 4 Recent Apps.



Map.



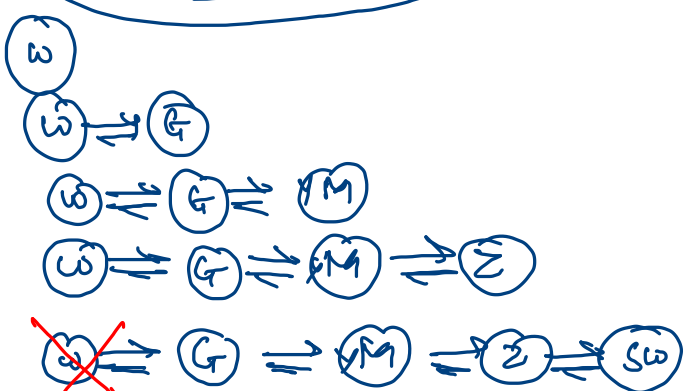
task → (1) WhatsApp → 45

(2) Google Map → 12

(3) Youtube Music → 5

(4) Zomato state -7

(5) Swiggy state 2.



(1) Youtube Music

↳ previous app re-open

↳ Idea

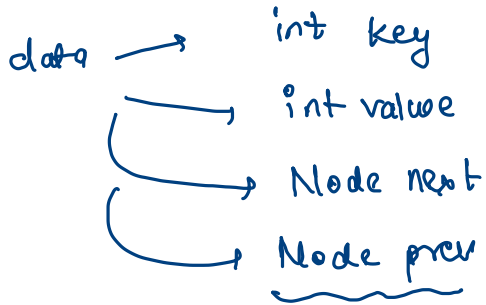
→ HashMap → AppName.

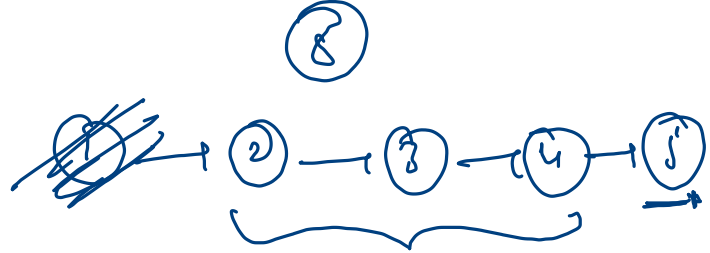
↳ Address of Node.

Key → App. Name.

Value → Node.

Structure of Node





Initialisation

- capacity
- Map

Map → Integer → key

Node → value

Get

put

Updation (map have key)

Insertion → if capacity available

(key is absent in map) → if capacity is full

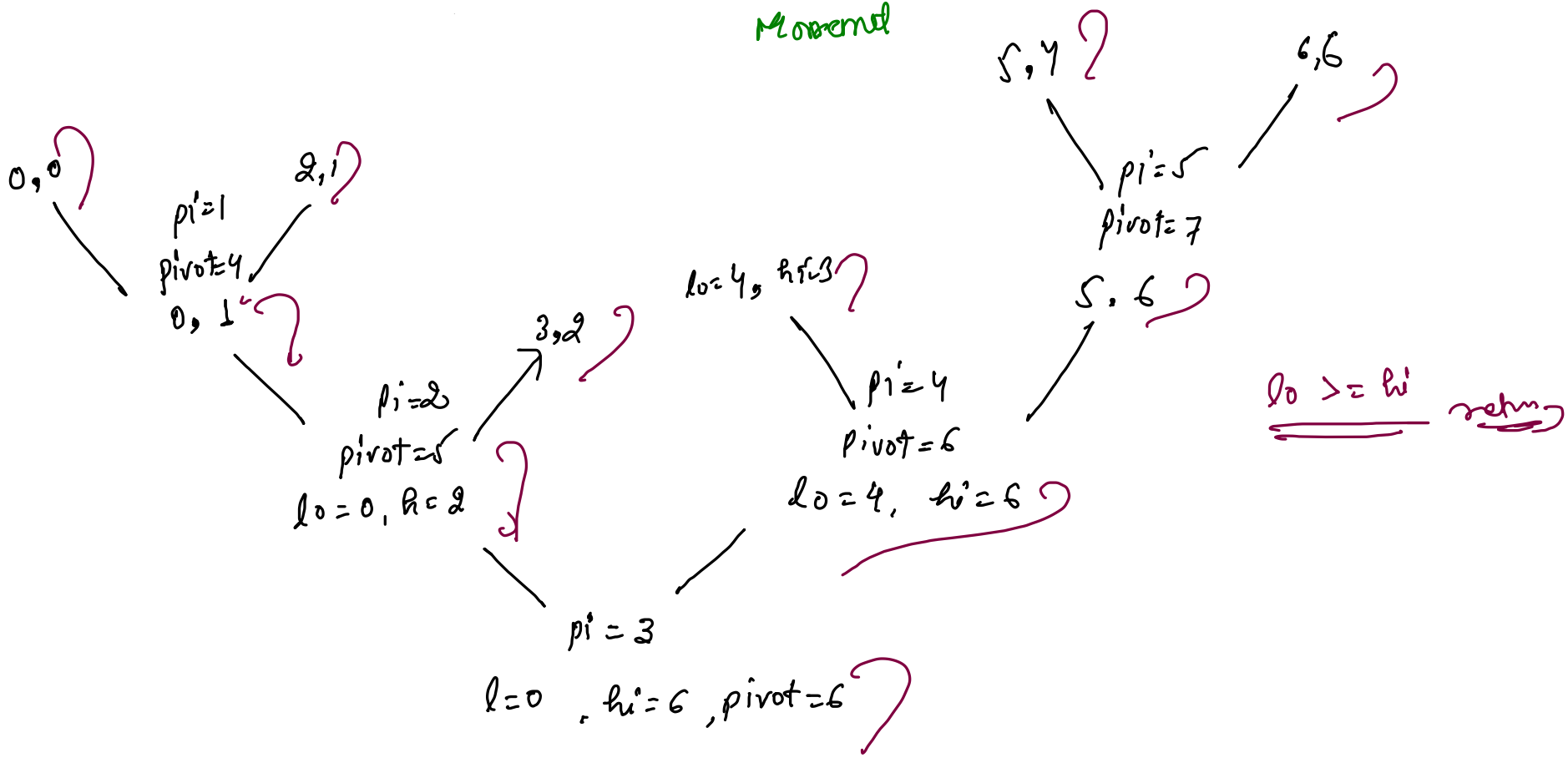
Quicksort In Array -

0	1	2	3	4	5	6
3	4	5	5	6	7	9

① Partition Array → Partitioning

② Quicksort
Recursive Movement

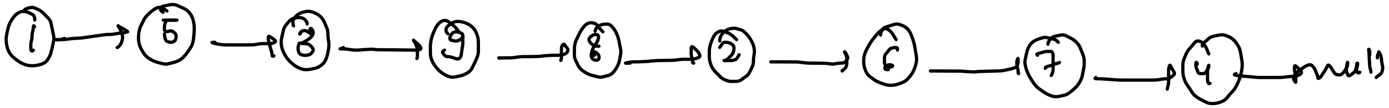
partition index
pivot = last



Single Linked List →

Quicksort in LinkedList :→

Partitioning in linked list is different than array.



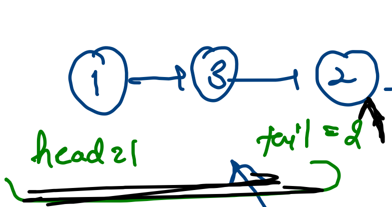
lo == head
hi == tail

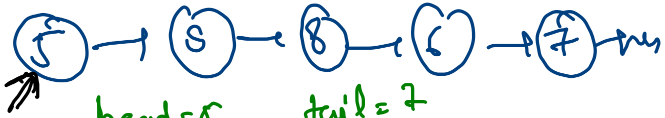
(i) Partition Node] Return type.

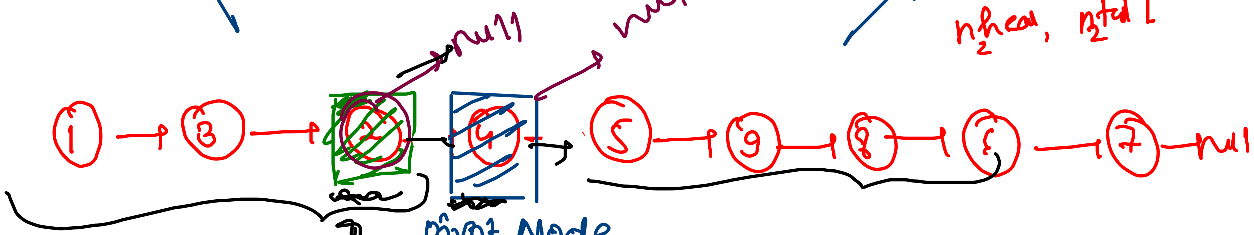
(2) Find Base case of Quicksort

(3) Edge case ↳ conditions.

[Return type of Quicksort function)



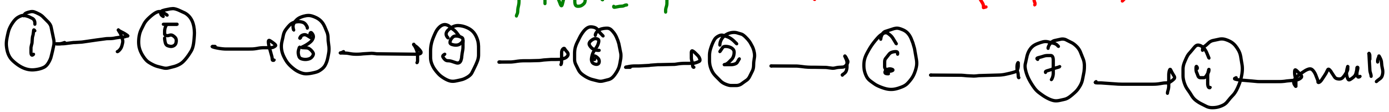


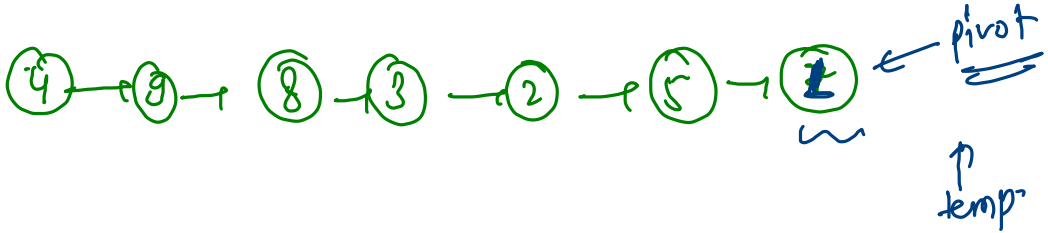


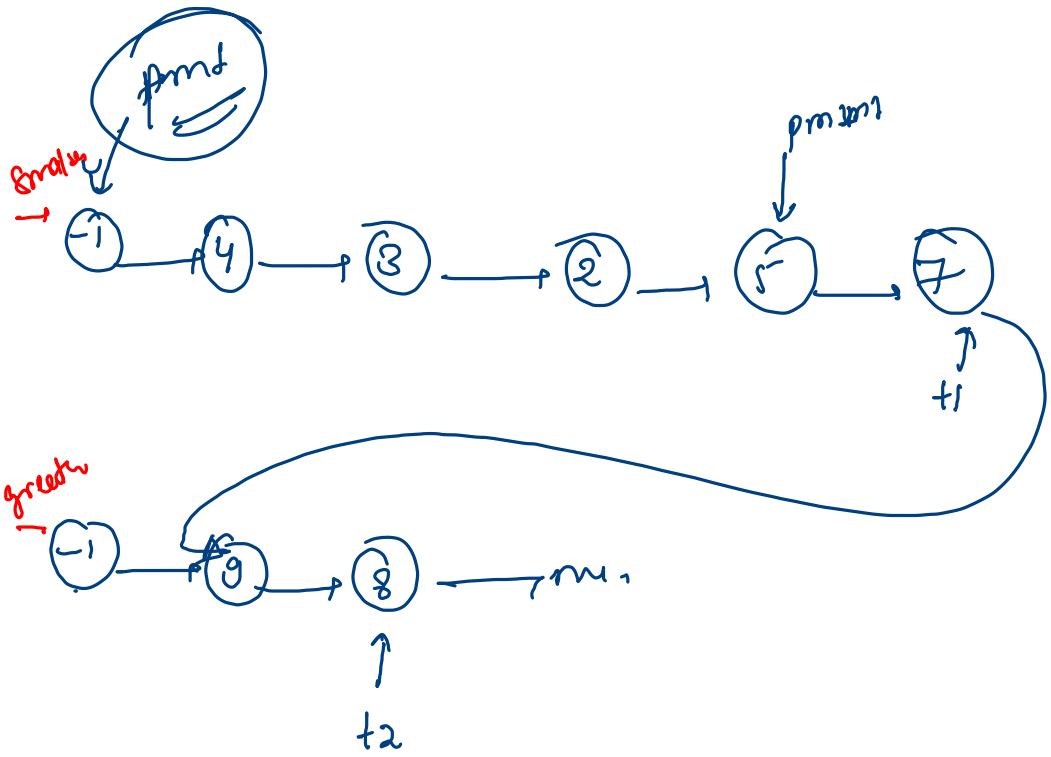
n1tail.next = pivotNode.
pivot.next = n2head.
return { n1head, n2tail }

Partition
├ Partition Node
└ Prime

(Partition Node minus 1

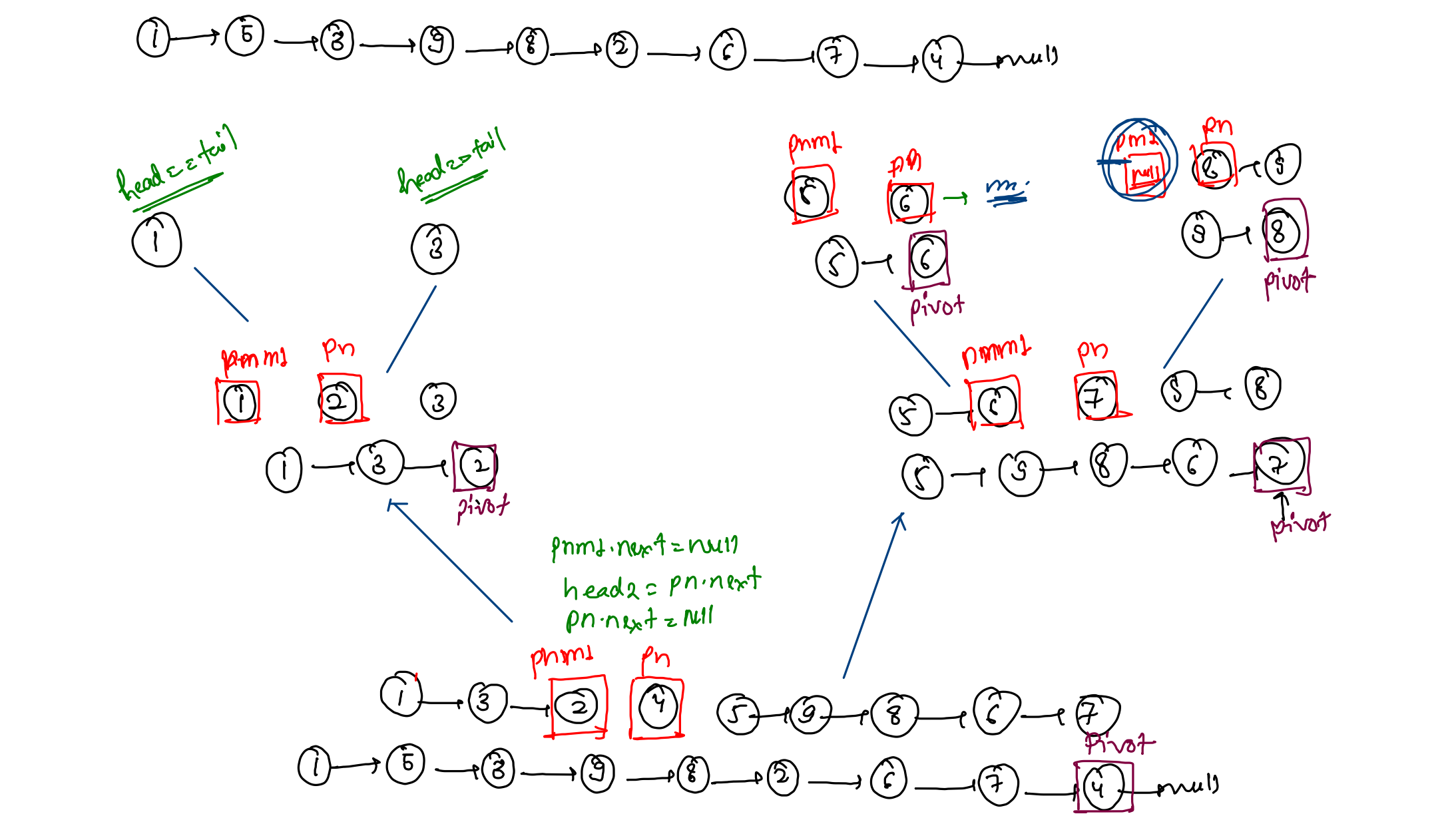







```java
public static PPair partition(ListNode head, ListNode pivot) {
    ListNode smaller = new ListNode(-1);
    ListNode greater = new ListNode(-1);
    ListNode t1 = smaller, t2 = greater, temp = head, pnm1 = null;

    while(temp != null) {
        if(temp.val <= pivot.val) {
            pnm1 = t1;
            t1.next = temp;
            t1 = temp;
        } else {
            t2.next = temp;
            t2 = temp;
        }
        temp = temp.next;
    }
    t2.next = null;
    t1.next = greater.next;
    return new PPair(pnm1 == smaller ? null : pnm1, pivot);
}
```


1 → 5 → 3 → 9 → 8 → 2 → 6 → 7 → 4 → null
head1 = tail
head2 = tail
pnm
pn
pivot
pnm.next = null
head2 = pn.next
pn.next = null
Pivot