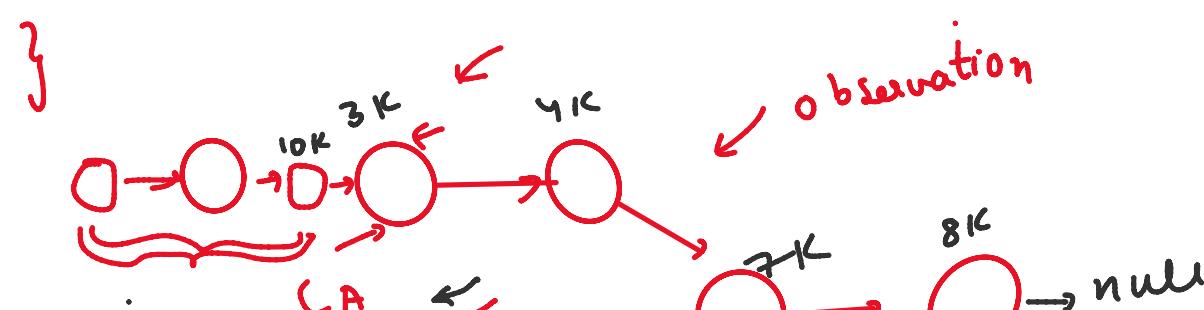
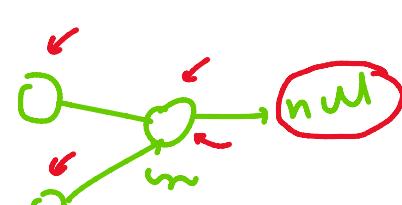
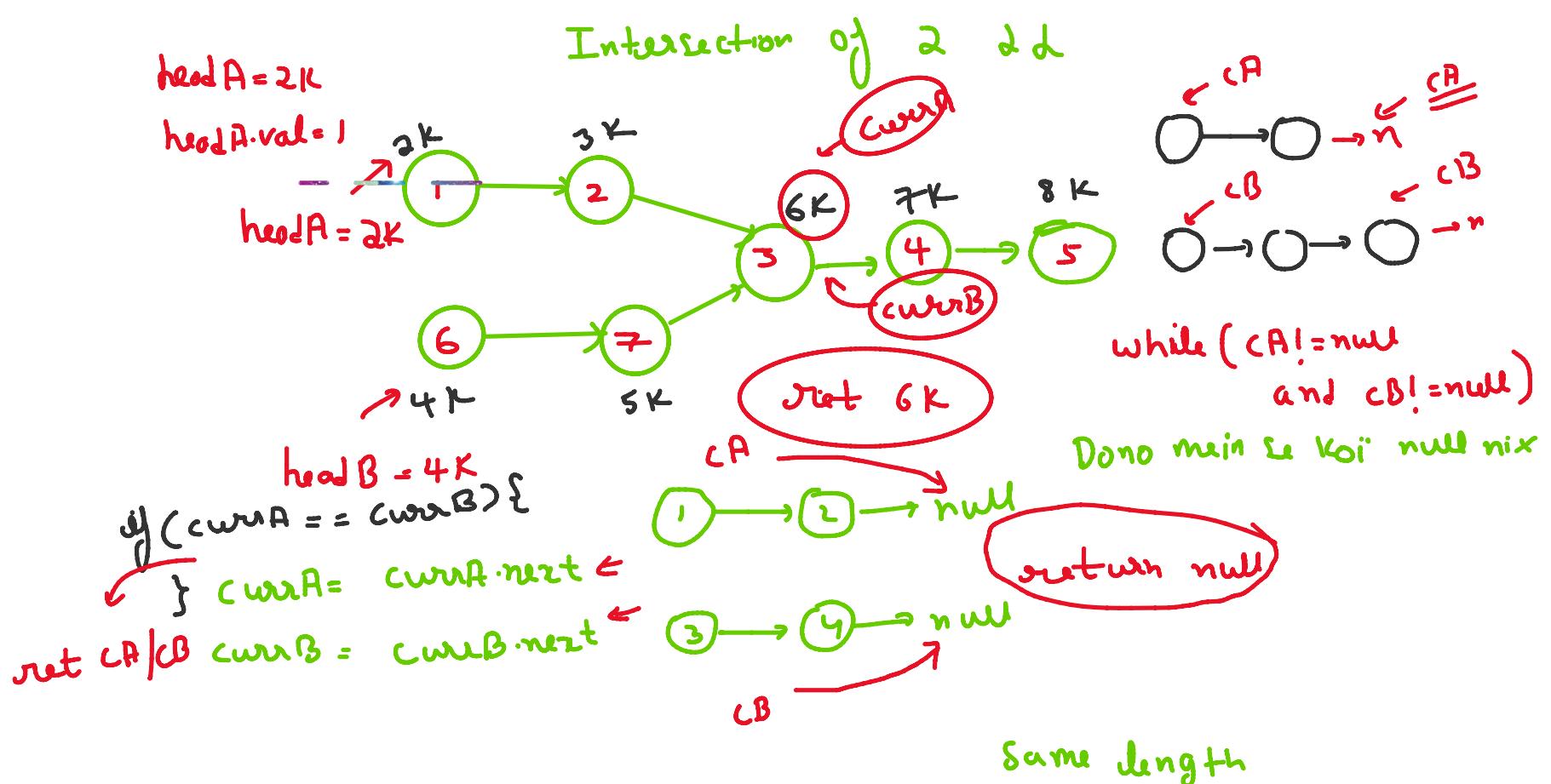
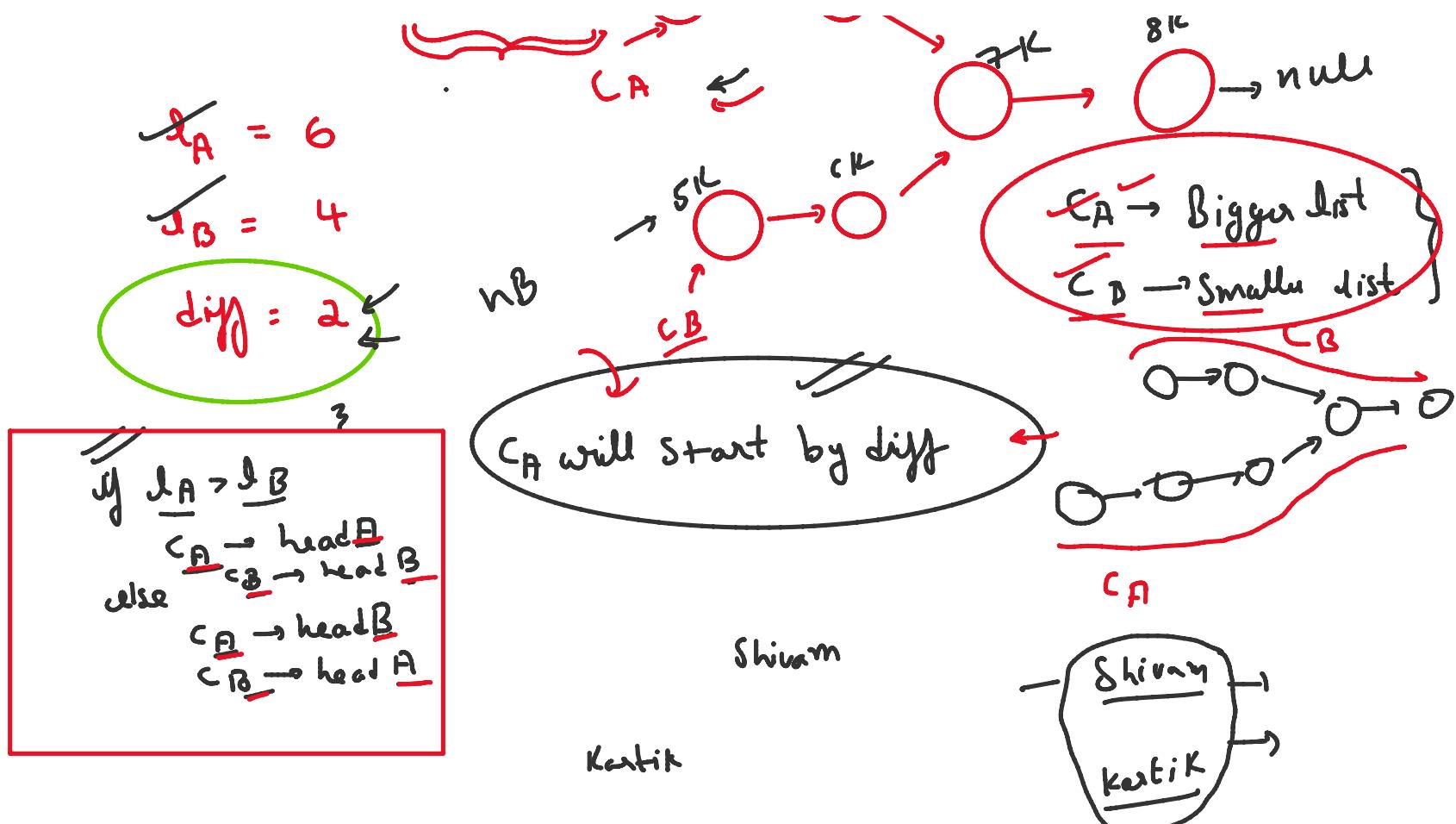


Questions to be covered

- 1) Intersection of 2 LL
- 2) Odd even LL
- 3) Delete without head LL

Optional -> sum of 2 LL





i) l_A and l_B calculate

ii) if ($l_A > l_B$)

$c_A \rightarrow H_A$

$c_B \rightarrow H_B$

diff $\rightarrow l_A - l_B$

else

$c_A \rightarrow H_B$

$c_B \rightarrow H_A$

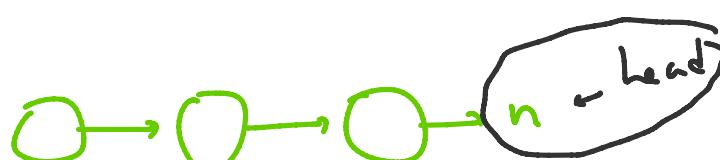
diff $\rightarrow l_B - l_A$

iii) starting point ka boost

while (diff != 0) {

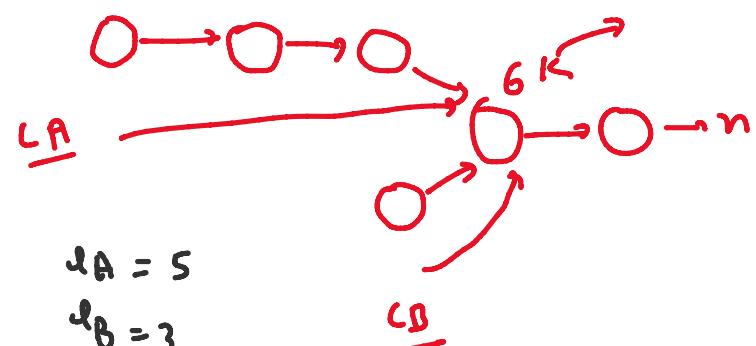
} $c_A \rightarrow c_A.next \rightarrow diff--$

iv) equal length vala code ↶



Count = 3

```
public ListNode getIntersectionNode(ListNode headA, ListNode headB) {
    int lA = lengthL(headA);
    int lB = lengthL(headB);
    ListNode cA = null;
    ListNode cB = null;
    int diff = 0;
    if (lA > lB) {
        cA = headA;
        cB = headB;
        diff = lA - lB;
    } else {
        cA = headB;
        cB = headA;
        diff = lB - lA;
    }
}
```

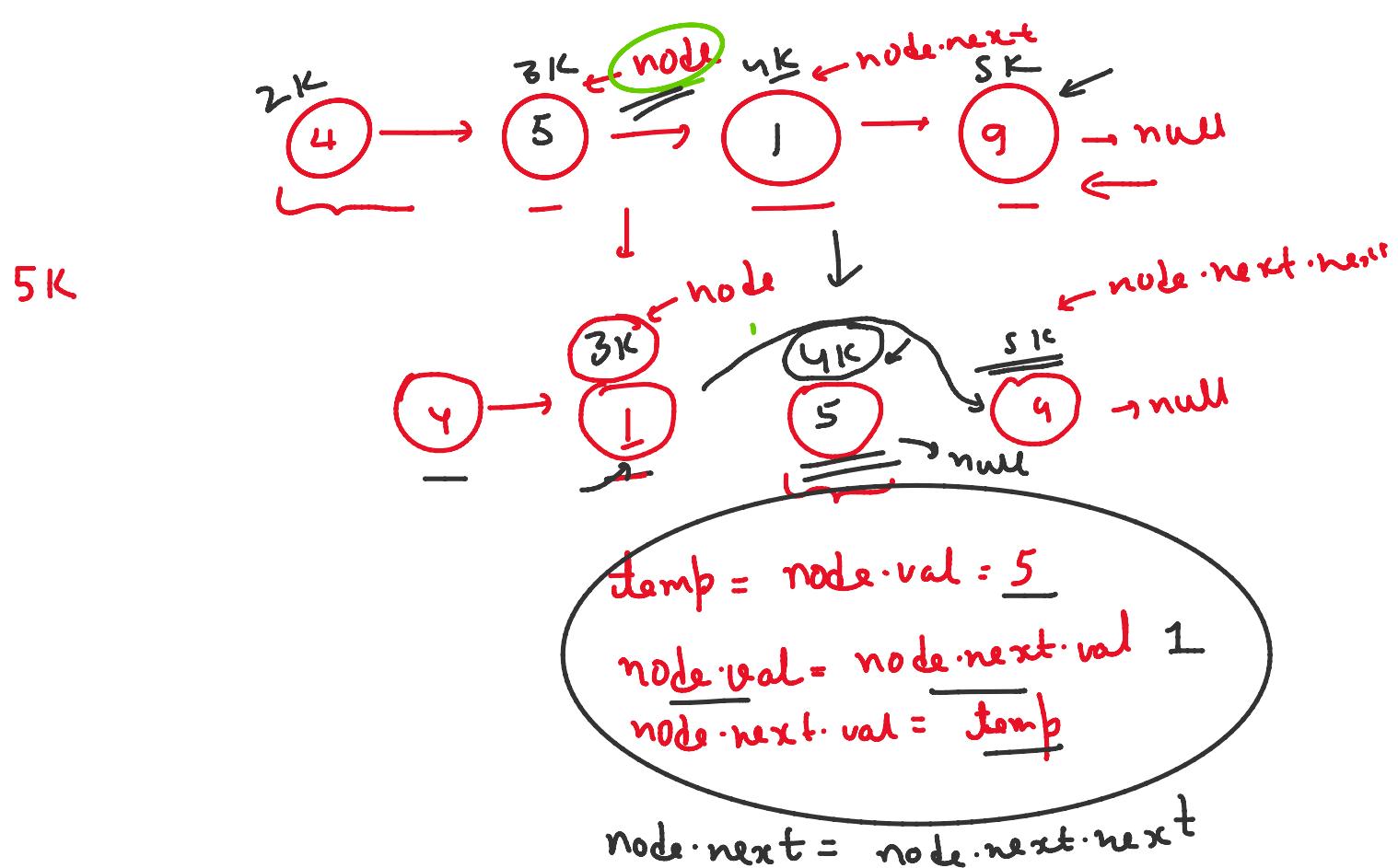


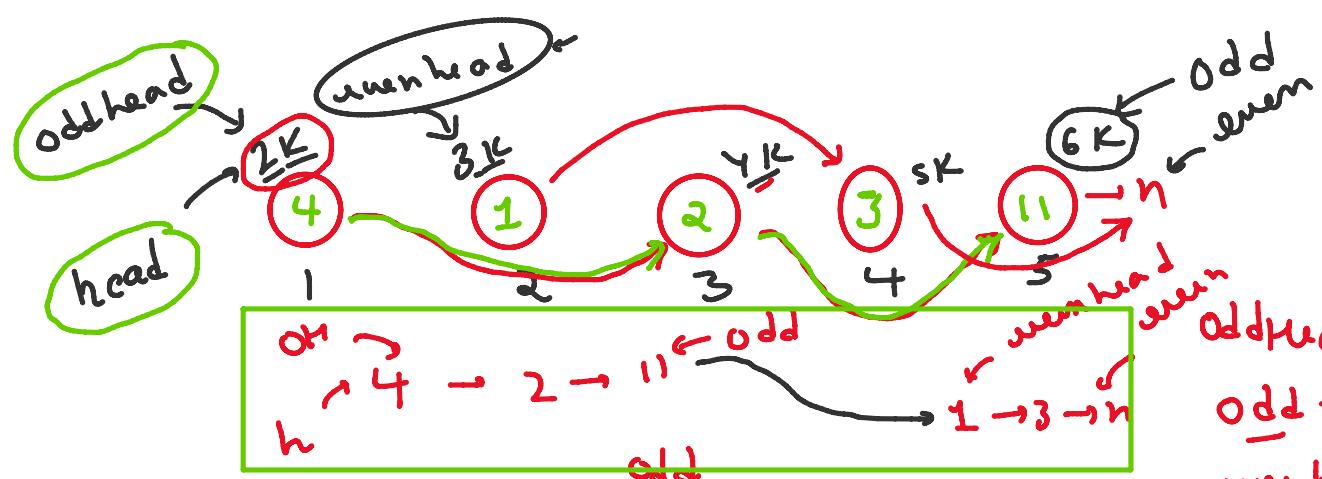
```

//starting point ko boost = diff
while(diff != 0) {
    cA = cA.next; ←
    diff--;
}
while(cA != null && cB != null) {
    if(cA == cB){ ←
        return cA;
    }
    cA = cA.next;
    cB = cB.next;
}
return null;
}

```

$$diff = 2 \neq 0$$





$4 \rightarrow 2 \rightarrow 11$

~~newList~~

$4 \rightarrow 2 \rightarrow 11 \rightarrow 1 \rightarrow 3 \rightarrow \text{null}$

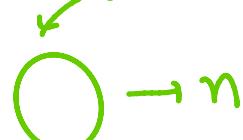
$211.\text{next} = 4K$

$$\left. \begin{array}{l} \underline{\text{odd.next}} = \underline{\text{odd.next.next}} \\ \underline{\text{odd}} = 4K (\underline{\text{odd.next}}) \end{array} \right\}$$

$\text{node} = \text{null}$

$\text{node}.\text{next} \rightarrow \text{npe}$

$\text{null} \xrightarrow{\text{odd}} \text{npe}$

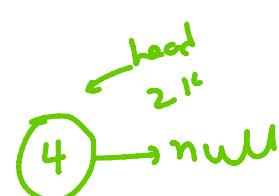


$\text{odd.next} = \text{null}$

$\text{odd.next.next} = \text{npe}$

$\text{head} = \text{null}$

```
ListNode oddHead = head;
ListNode odd = head;
ListNode evenHead = head.next; n
```



```

ListNode oddHead = head;
ListNode odd = head;
ListNode evenHead = head.next;
ListNode even = head.next;
while(odd != null && odd.next != null && even != null && even.next != null) {
    odd.next = odd.next.next;
    odd = odd.next;
    even.next = even.next.next;
    even = even.next;
}
odd.next = evenHead;
return oddHead;
}

```

