C, C++, DSA in depth

Circular linked list

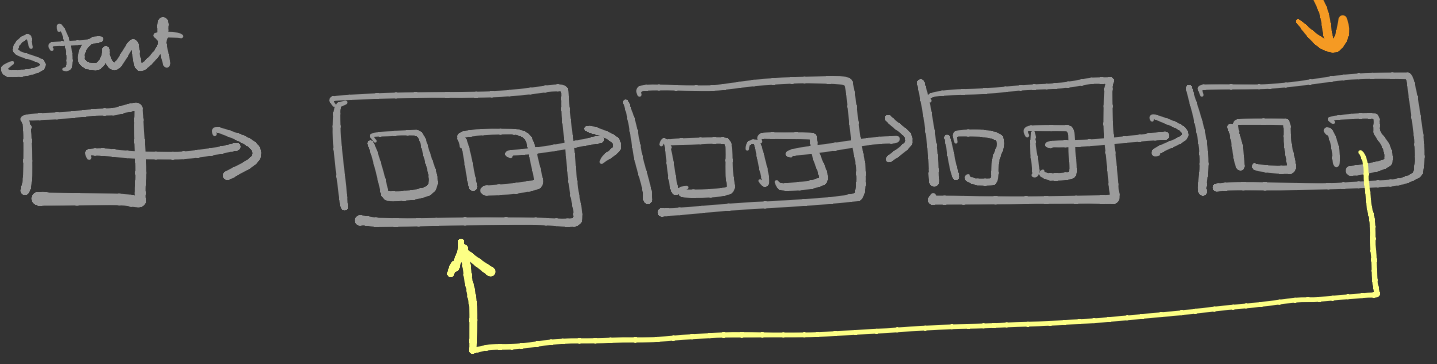
Saurabh Shukla (MySirG)

Agenda

1. Not utilizing next pointer of last node in SLL
2. Circular linked list
3. Comparison of CLL with SLL
4. Small change big difference
5. Insertion and Deletion in CLL

Not utilizing next pointer of last node in SLL

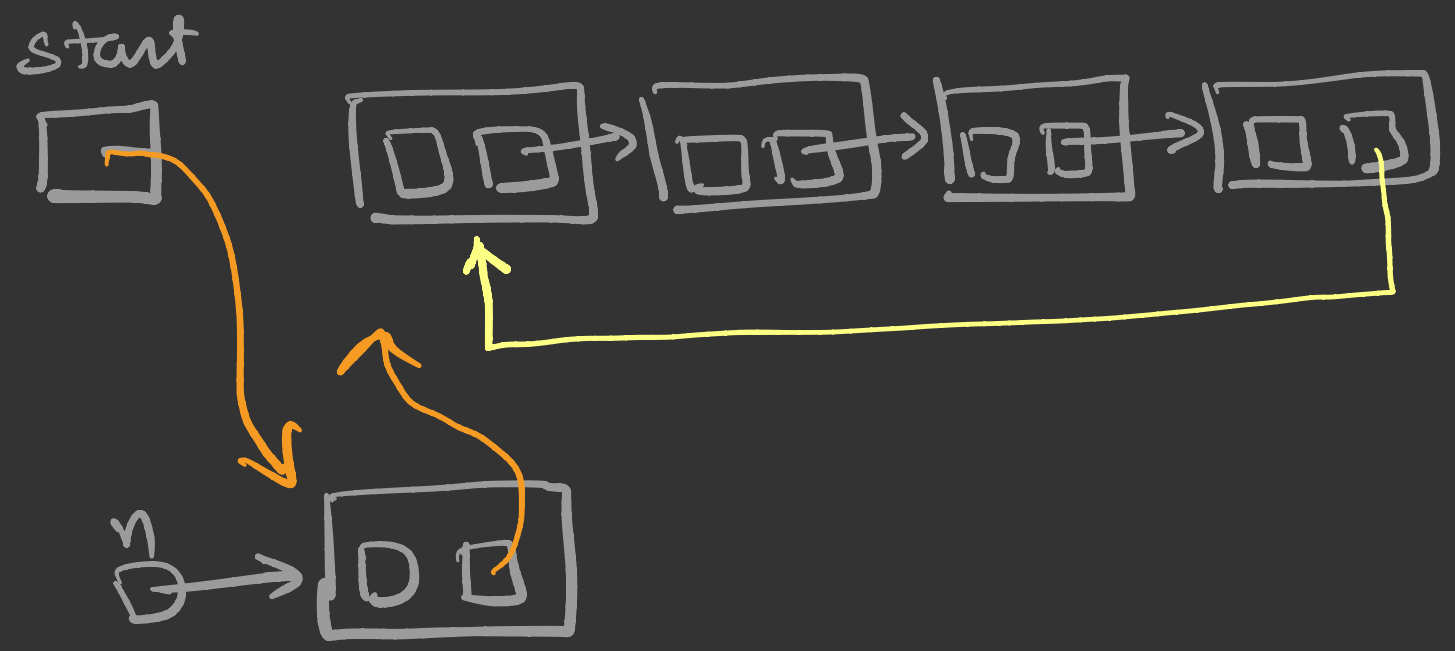
Solution



if(t → next==start)

t last node को point kar raha hai

Circular linked list

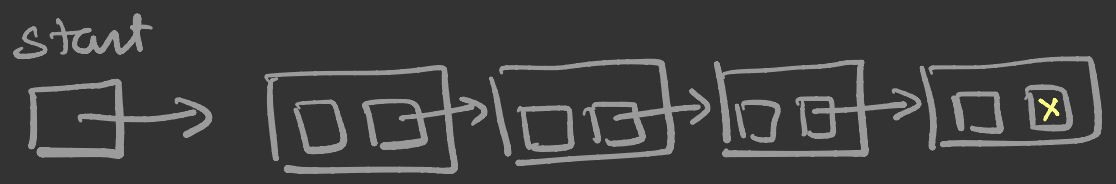



```
n->item=data;
n->next = start;
start = n;
```

traversing

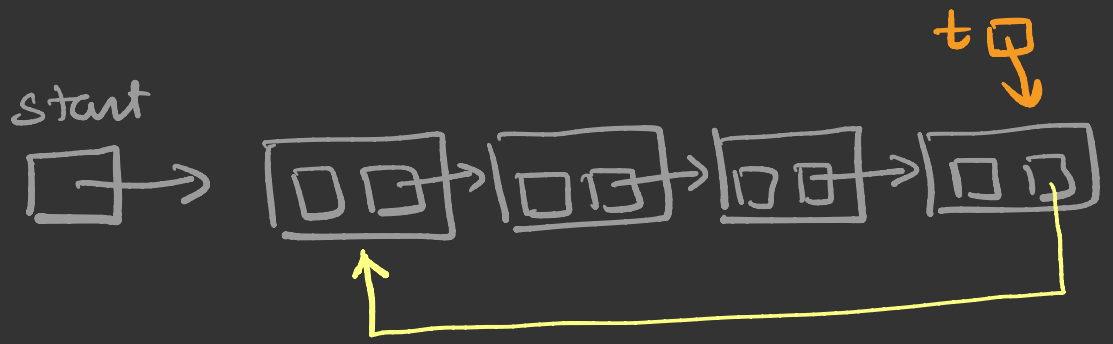
```
t->next = n;
```

Comparison of CLL with SLL



Insertion
1. First — no traversing
2. Last — traversing

Deletion
1. First — no traversing
2. Last — traversing



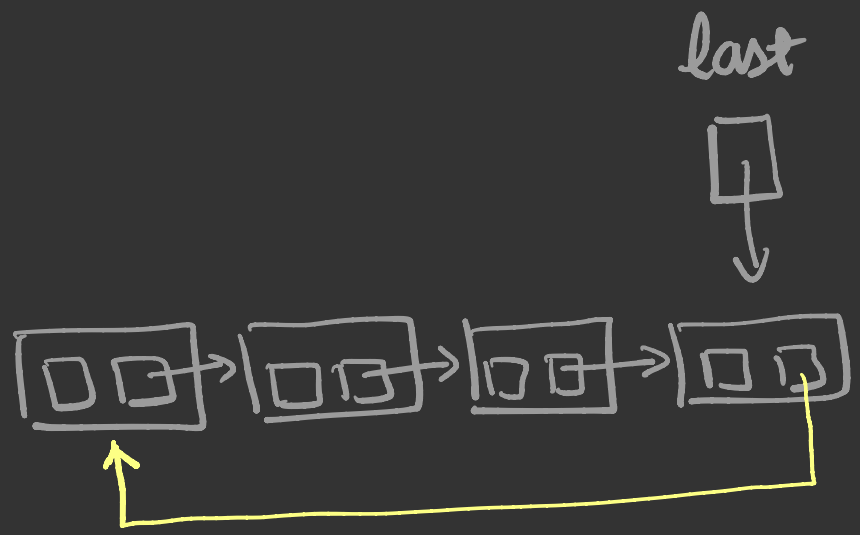
Insertion
1. First — traversing
2. Last — traversing

Deletion
1. First — traversing
2. Last — traversing.

Small Change, Big Difference

Insertion
1. First no traversing
2. Last traversing

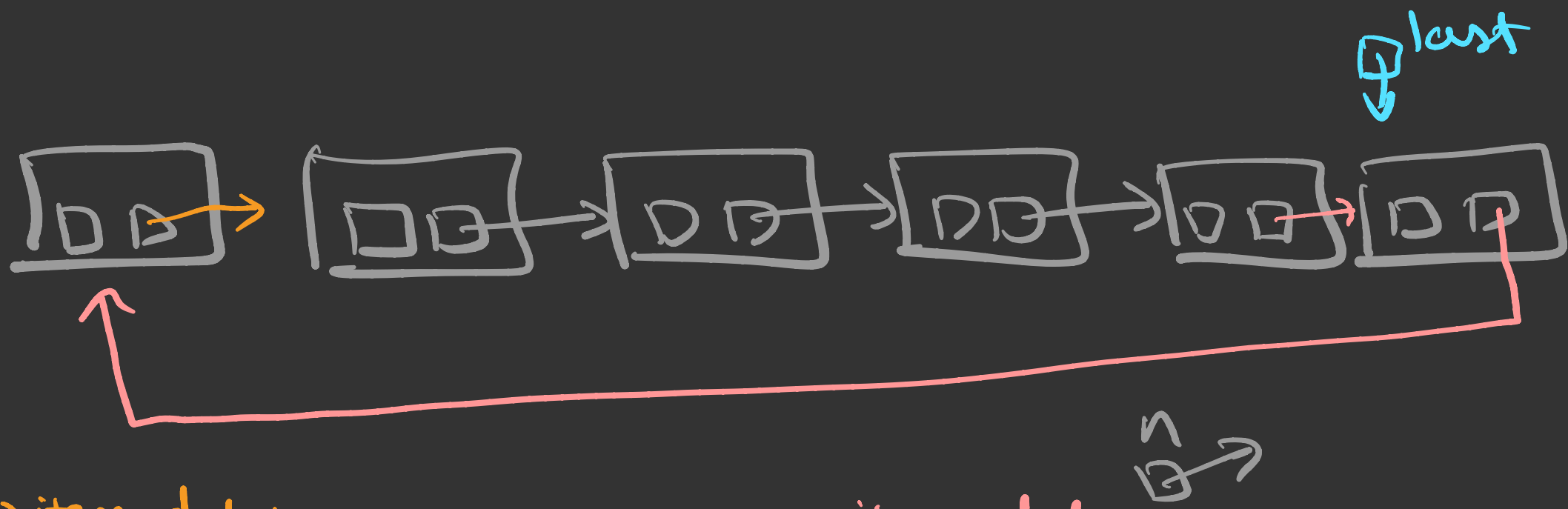
Deletion
1. First no traversing
2. Last traversing



Insertion
1. First no traversing
2. Last no traversing

Deletion
1. First no traversing
2. Last traversing

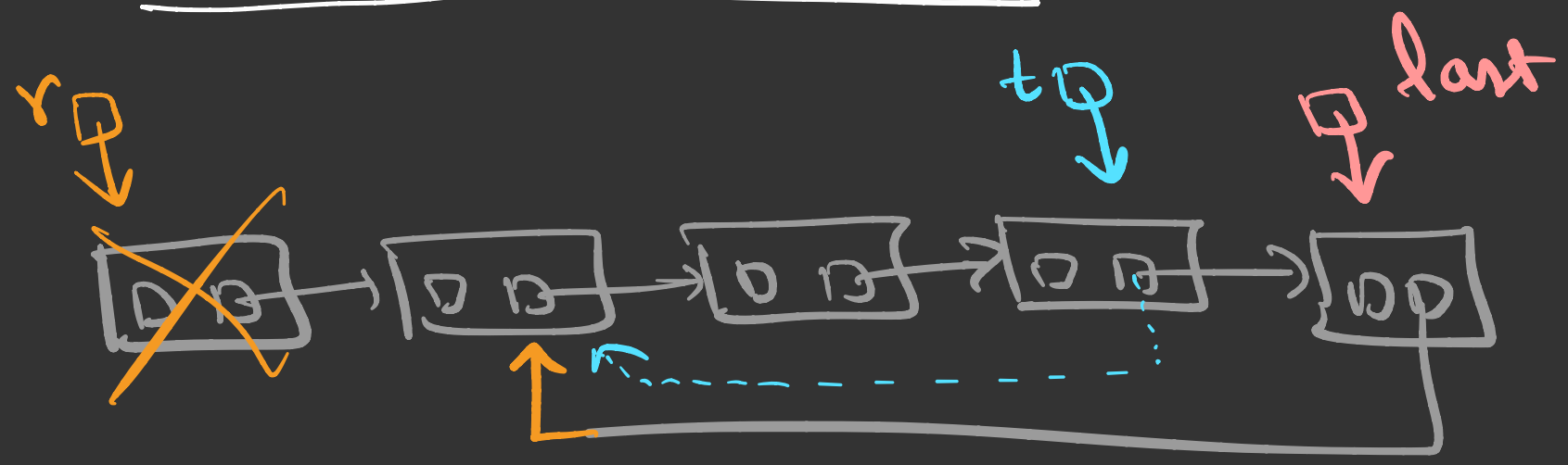
Insertion in CLL




```
n->item=data;
n->next= last->next
last->next=n;
```

```
n->item=data;
n->next= last->next;
last->next = n;
last=n;
```

Deletion in CLL




```
r = last->next;
last->next = r->next;
delete r;
```

```
t->next = last->next;
delete last;
last = t;
```