DSA through C++

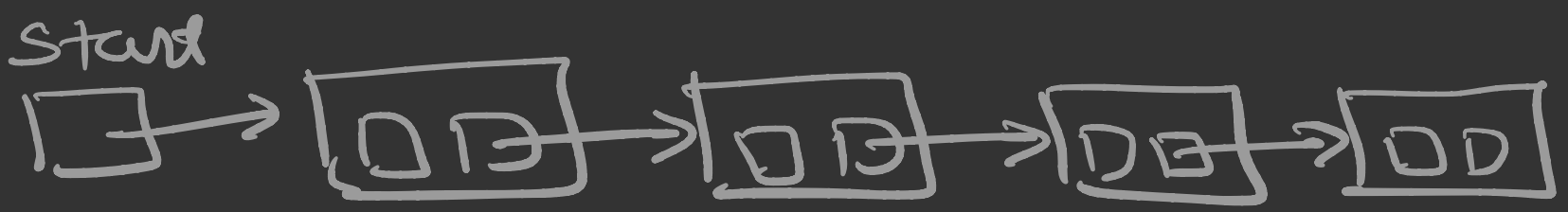
Doubly linked list

Saurabh Shukla (MySirG)

Agenda

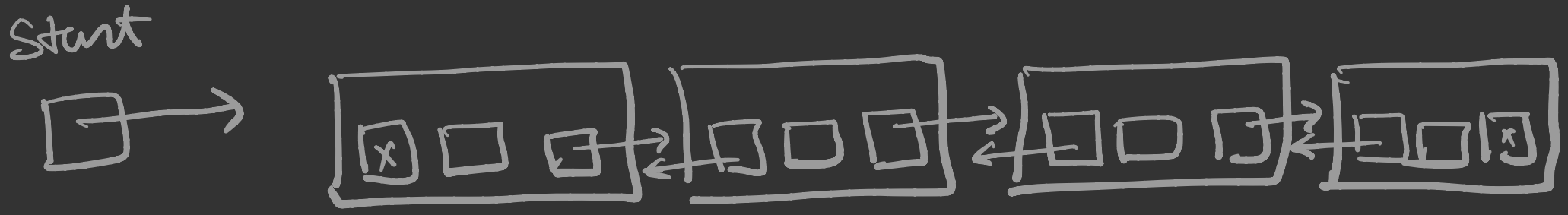
1. Shortcomings of singly linked list
2. Doubly linked list
3. node
4. insertion
5. deletion

Shortcomings of Singly linked list

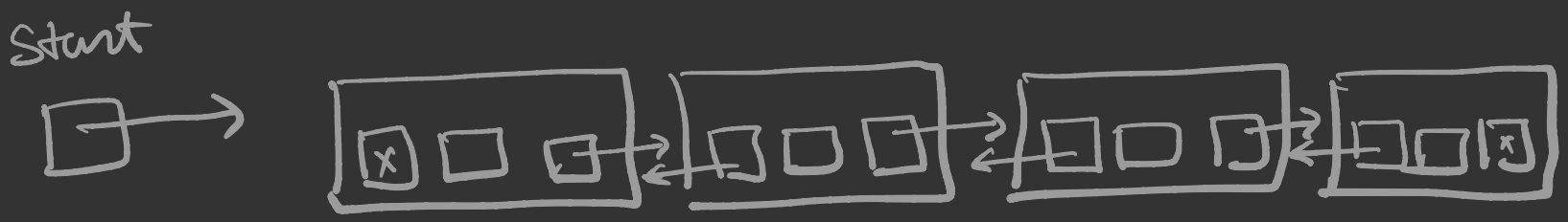


In SLL, you can move only in the forward direction

Doubly linked list

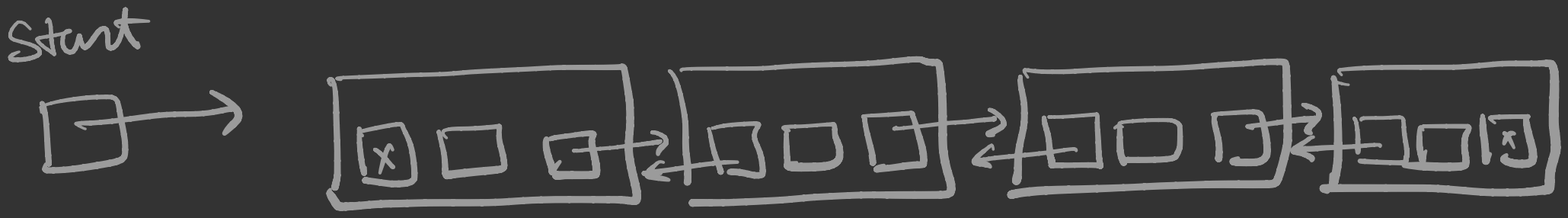


node




```
struct node
{
    node *prev;
    int item;
    node *next;
};
```

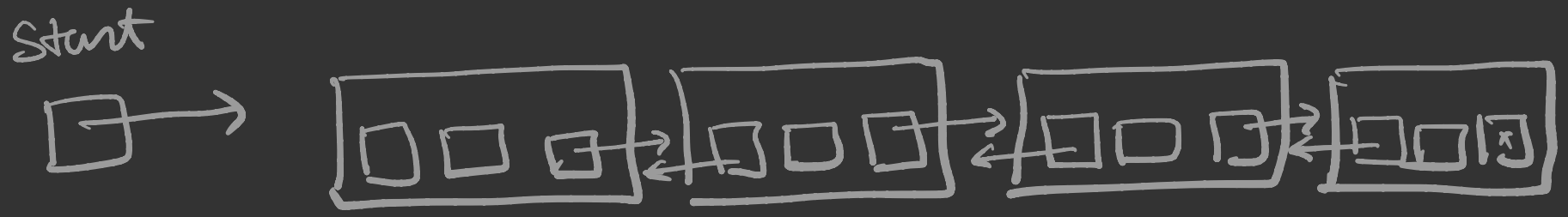
Insertion

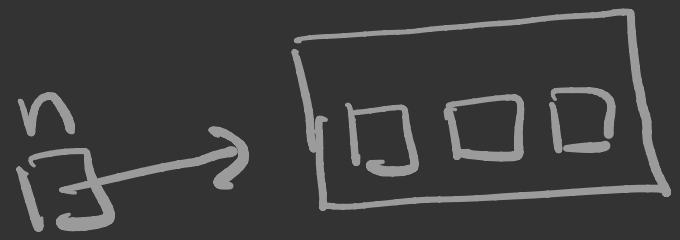


Insertion

1. At First
2. At Last
3. After a node

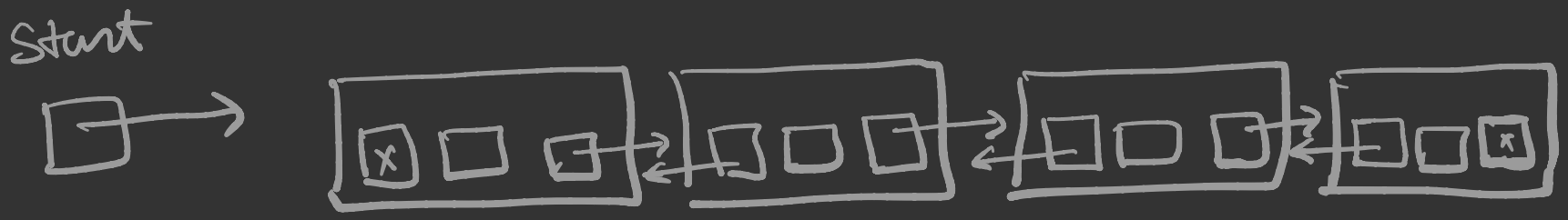
Insert as a first node





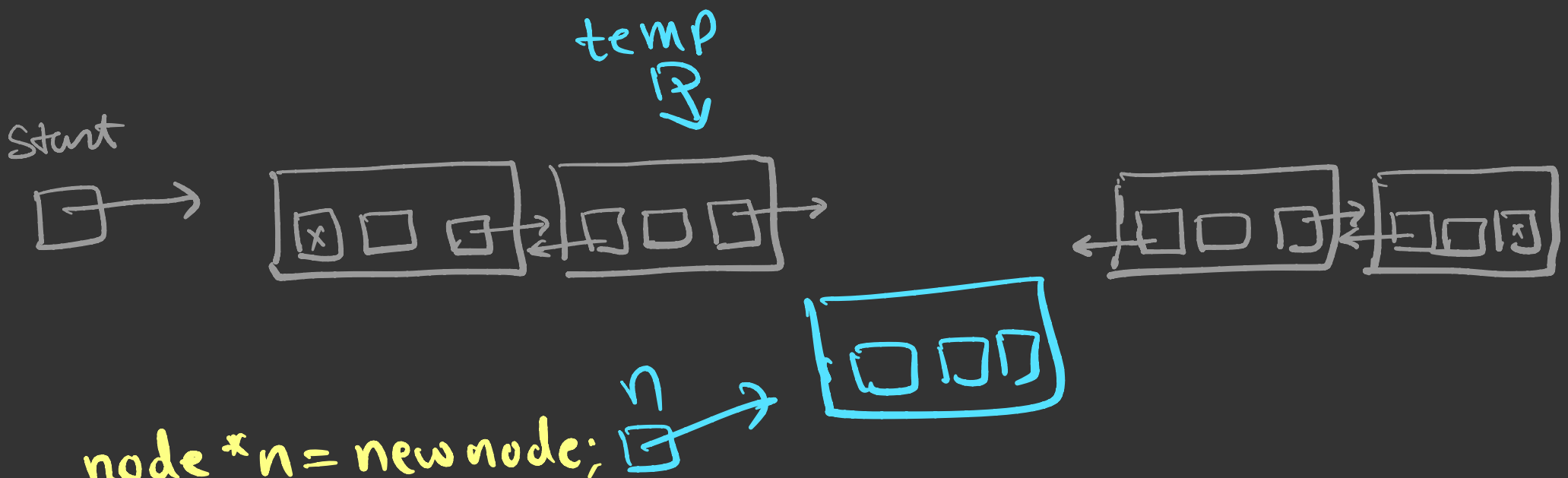

```
node *n = new node;
n -> item = data;
n -> prev = NULL;
n -> next = start;
start -> prev = n;

start = n;
```

Insert at last




```
node *n = new node;
n -> item = data;
n -> next = NULL
t = start;
while (t -> next != NULL)
      t = t -> next;
n -> prev = t;
t -> next = n;
```

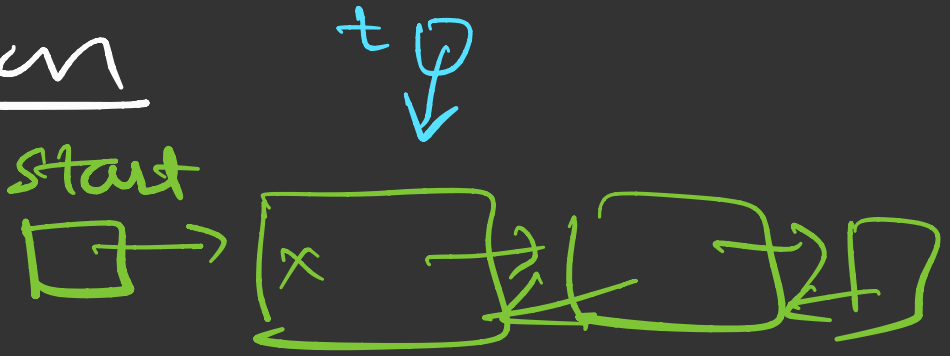
Insert After a node

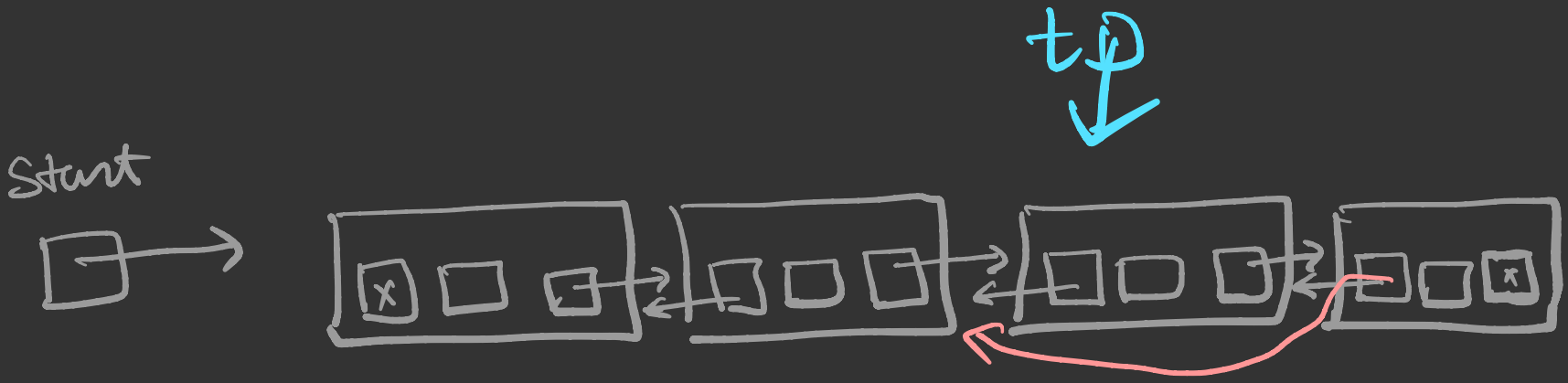



```
node *n= new node;
n->item = data;
n->prev = t;
n->next = t->next;
t->next->prev = n;
t->next = n;
```

Deletion

1. Delete First Node
2. Delete Last Node
3. Delete Specific node






```
t->next->prev = t->prev;
```

```
t->prev->next = t->next;
```