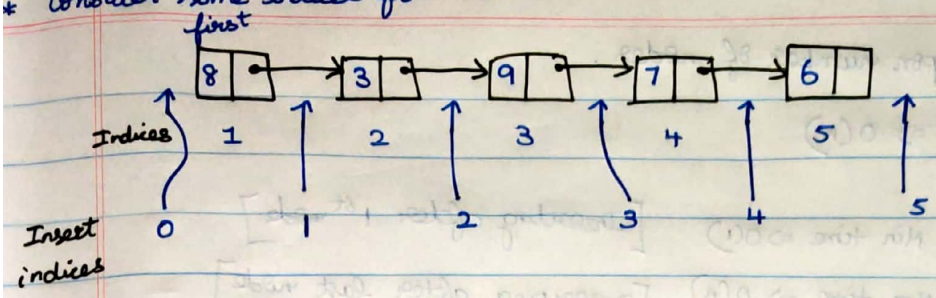
Inserting a linked list

* Consider some indices for the lists



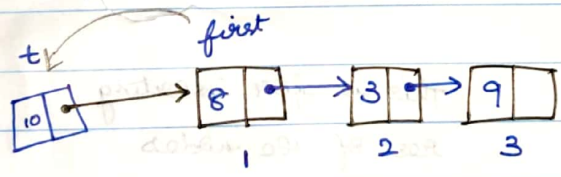
Cases

- Inserting before first node
- Inserting after a given position

Inserting a new node before first node :

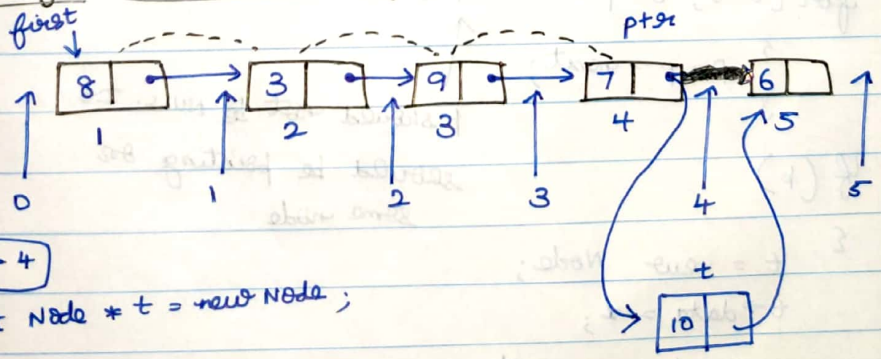
```
struct Node *t = new Node;
t->data = 10;
t->next = first;
first = t;
```

i) Create a new node.
ii) Insert the data.
iii) Link it with first node.
iv) Then point first to the new node



Time => O(1)

Inserting a new node at a given position :



pos = 4

```
struct Node *t = new Node;
t->data = 10;
p = first;
for (i=0; i<pos-1; i++)
    {p = p->next;
    }
t->next = p->next;
p->next = t;
```

i) Create a new Node,
ii) A pointer is required to make the link. Already t is there, so bring one pointer from the beginning.

* All the operations take constant time, but major time consuming is that bringing the pointer to (position-1)th place.

* It depends upon number of nodes.

* Hence, time => O(n)

↳ Min time => O(1) [Inserting after 1st node]

↳ Max time => O(n) [Inserting after last node]

Combining the two cases:

```
void insert (int pos, int x)
{
    struct Node *t, *p;
    if (pos == 0)
    {
        t = new Node;
        t->data = x;
        t->next = first;
        first = t;
    }
    else if (pos > 0)
    {
        p = first;
        for (i=0; i < pos-1 && p; i++)
        {
            p = p->next;
        }
        if (p)
        {
            t = new Node;
            t->data = x;
            t->next = p->next;
            p->next = t;
        }
    }
```

To create a new node

For traversing from first

This is inserting before the 1st node.

This is for inserting rest of the nodes

p should not be NULL. It should be pointing to some node