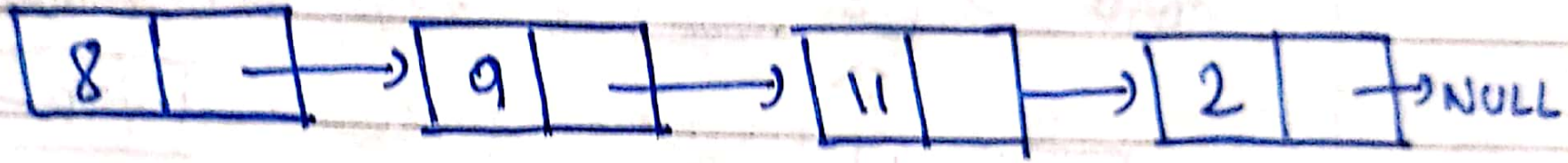
Insertion of a Node in a Linked list:

→ Consider the following linked list:



→ For insertion, we would first need to create that extra node. And

Date

then, we overwrite the current connection and make new connections. And that is how we insert a new node at our desired place.

→ Syntax for creating a node:

```
STRUCT NODE *PTR = (STRUCT NODE*) MALLOC(
                    SIZEOF(STRUCT NODE))
```

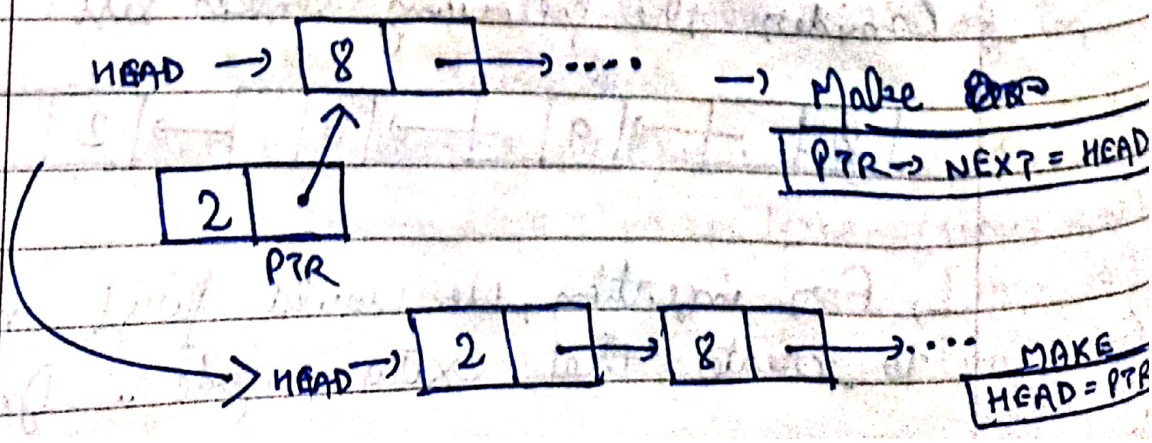
→ This above syntax will create a node, and the next thing one would need to do is set the data for this node.

```
PTR→DATA = a;
```

→ This will set the data.

→ CASE 1: Insert at the beginning:

→ In order to insert the new node at the beginning. We would need to have the head pointer pointing to this new node and the new node's pointer to the current head.



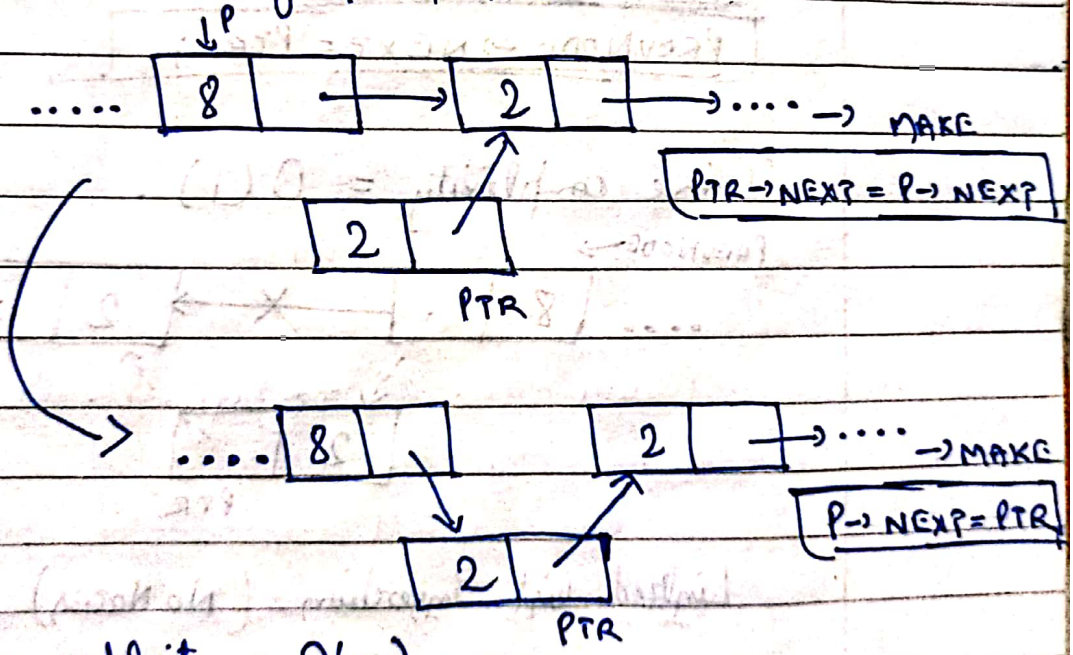
→ Time complexity => O(1).

→ CASE 2: Insert in between:

→ Assuming index starts from 0, we can insert an element at index i > 0 as follows:

1) Bring a temporary pointer P pointing to the node before the element you want to insert in the linked list.

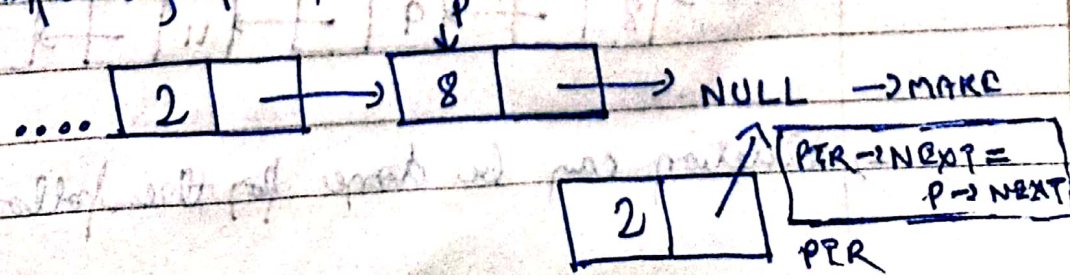
2) Since we want to insert b/w 8 & 2, we bring pointer P to 8.

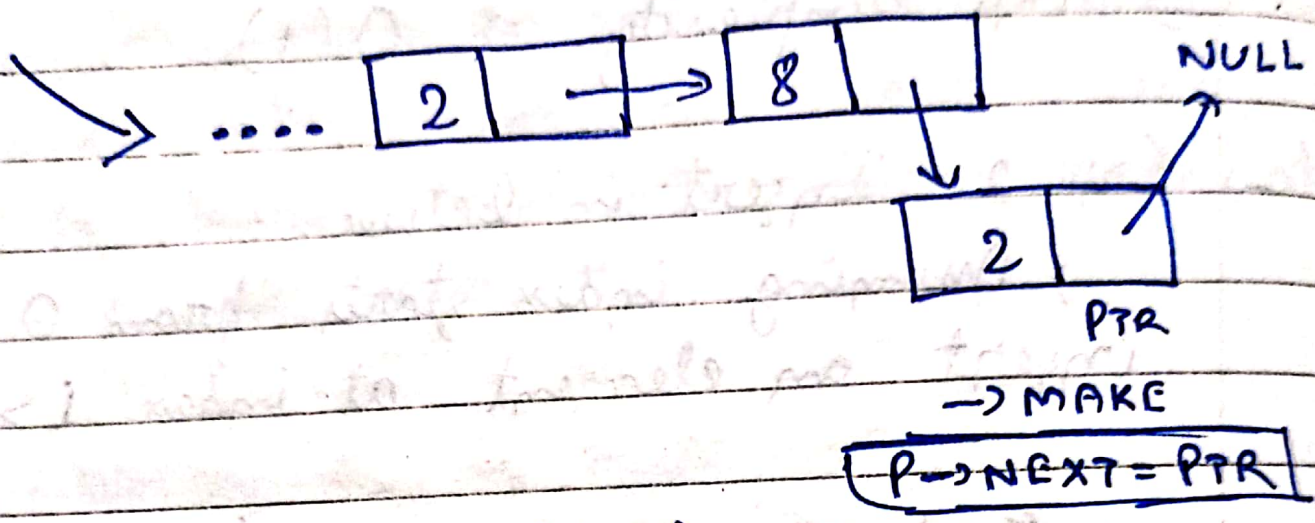


→ Time complexity = O(n).

→ CASE 3: Insert at the end:

→ In order to insert an element at the end of the linked list, we bring a temporary pointer to the last element.





→ MAKE

P→NEXT = PTR

→ Time complexity = O(n).

→ CASE 4: Insert after a node:

→ Similar to the other cases, PTR can be inserted after a node as follows: (PREVNODE'S ADD. IS GIVEN IN IT).

PTR→NEXT = PREVNODE→NEXT;

PREVNODE→NEXT = PTR;

→ Time complexity = O(1).