LINKED LIST

LINKED LIST

Why Linked List?

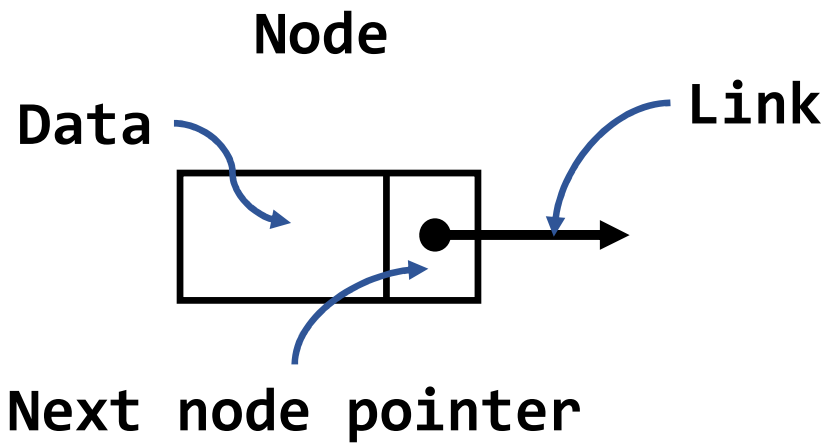
ARRAY V/S LINKED LIST

Arrays	Linked List
Maximum capacity of the array should be known beforehand.	Memory can be changed at the time of usage according to our need.
Insertion and deletion is costlier, means we have to swap other elements.	Insertion and deletion is very easy in linked list.

LINKED LIST

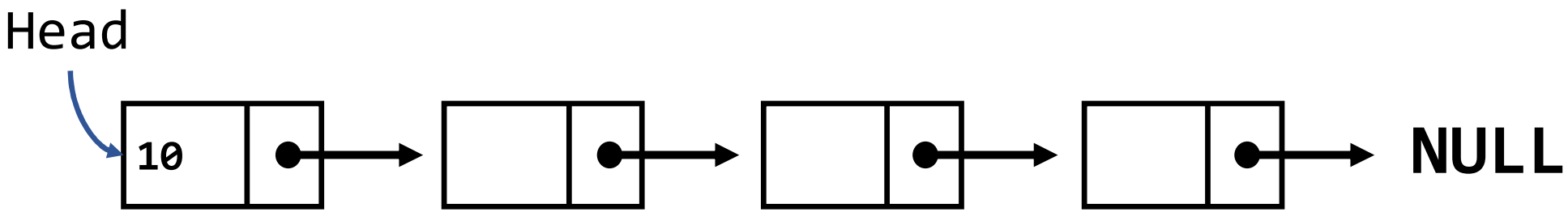
What is a Linked List?

Linked Lists are similar to arrays but they are not stored in continuous manner in memory like arrays and their respective nodes are connected by using pointers.



LINKED LIST

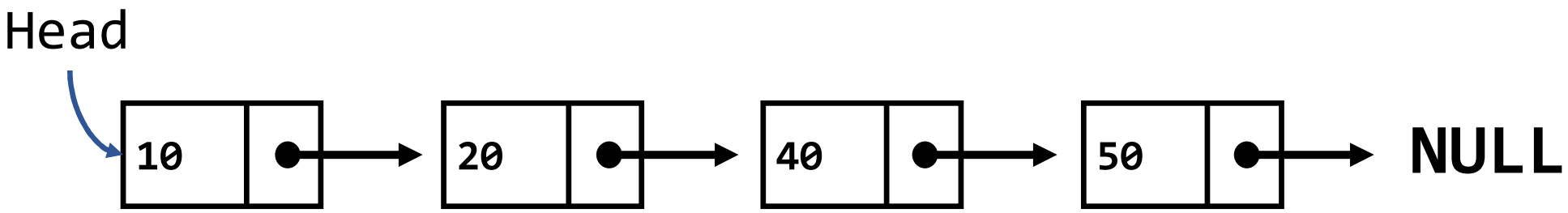
A complete Linked List looks something like this.



LINKED LIST

Insertion in Linked List:-

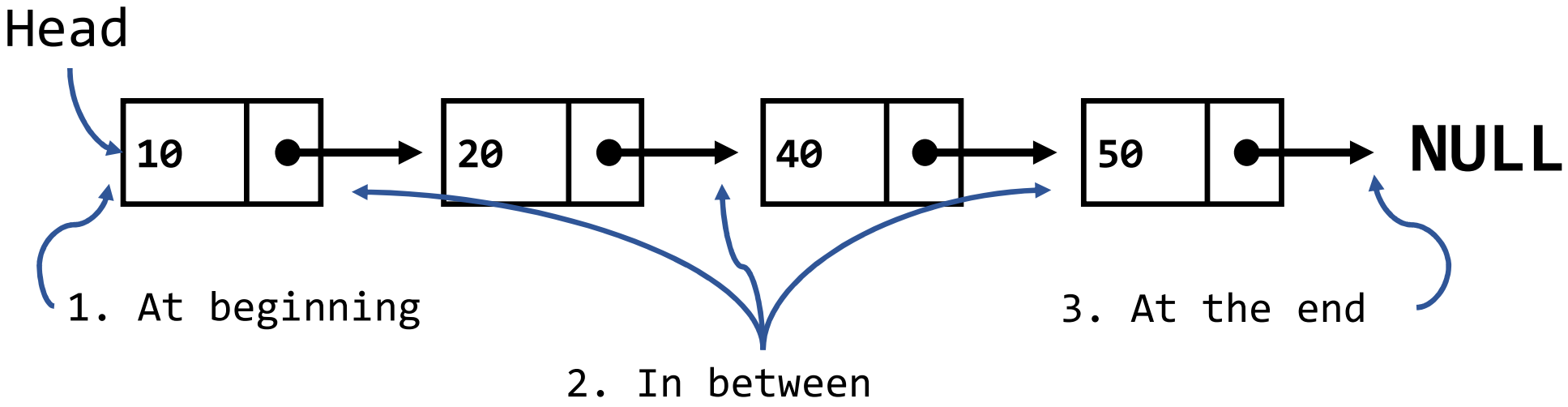
Let’s take this as a linked list



LINKED LIST

Insertion in Linked List:-

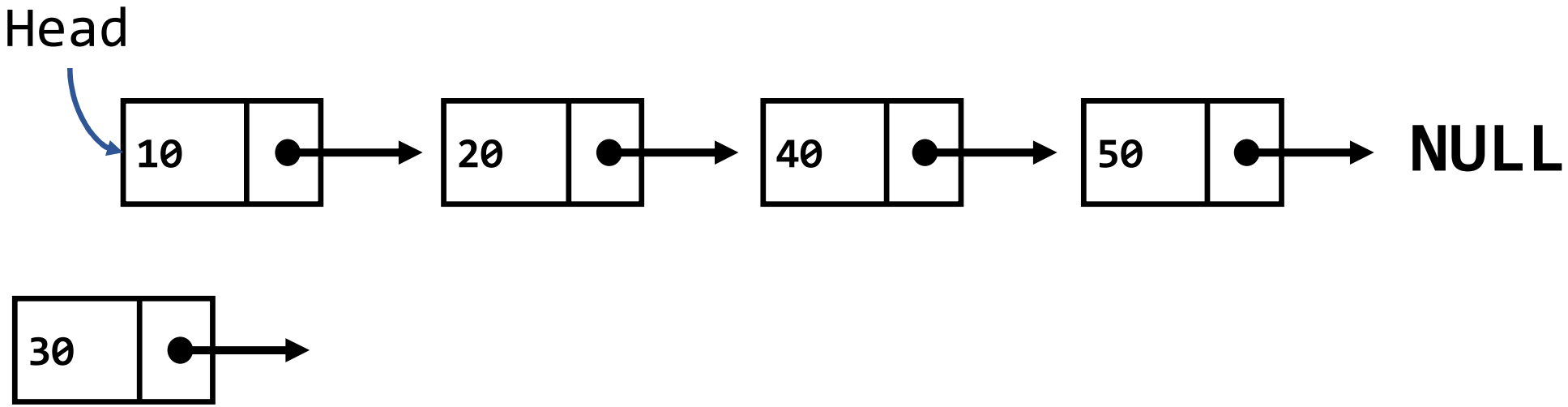
We have three options to insert



LINKED LIST

Insertion in Linked List:-

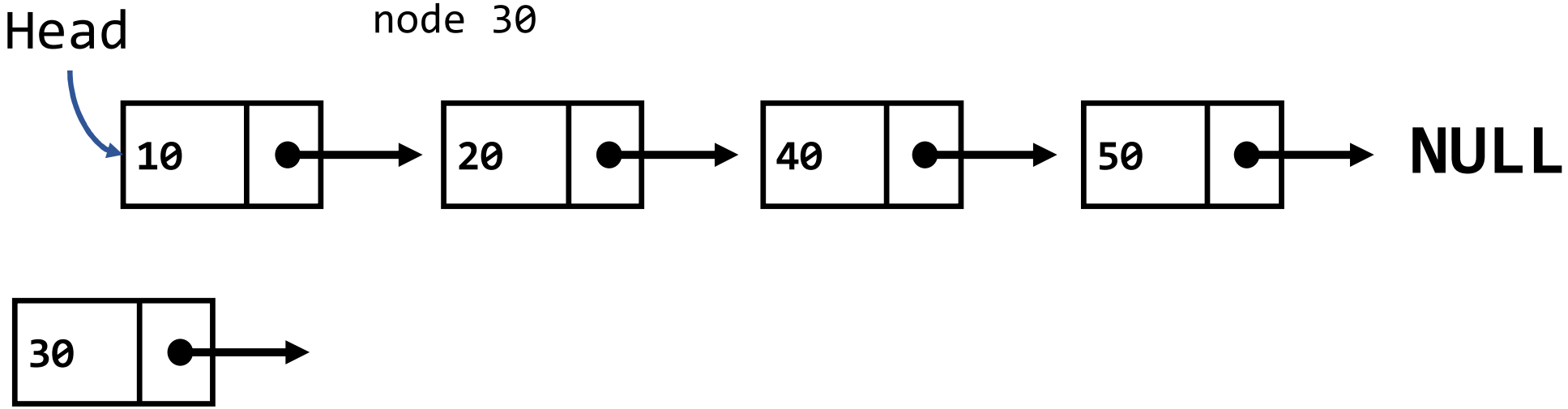
Let’s say we want to insert a node
with value 30 in the beginning



LINKED LIST

Insertion in Linked List:-

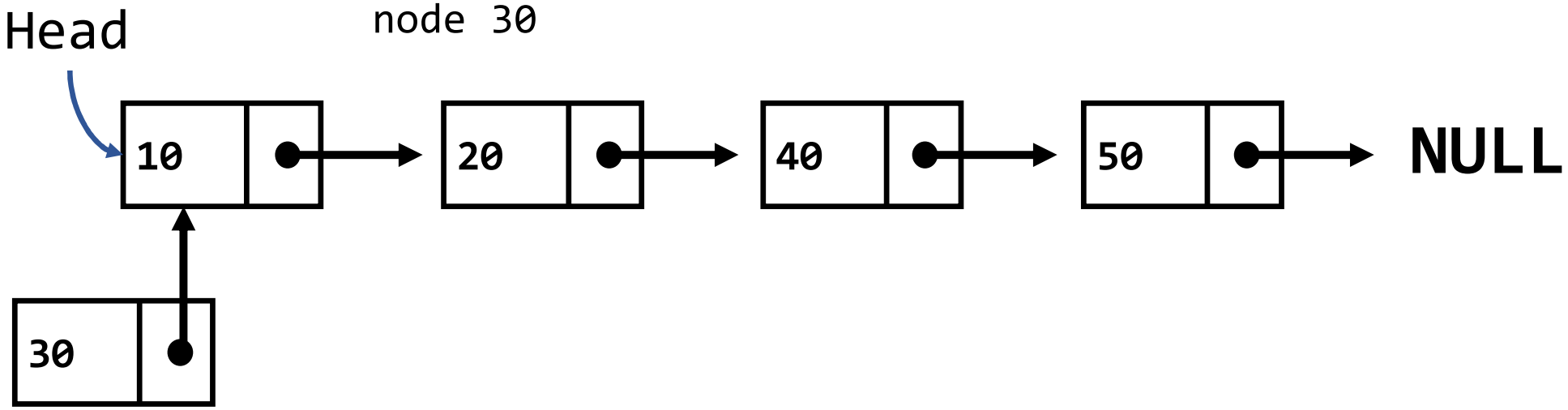
So we can simply make a link from 30 to 10 and change Head pointer to node 30



LINKED LIST

Insertion in Linked List:-

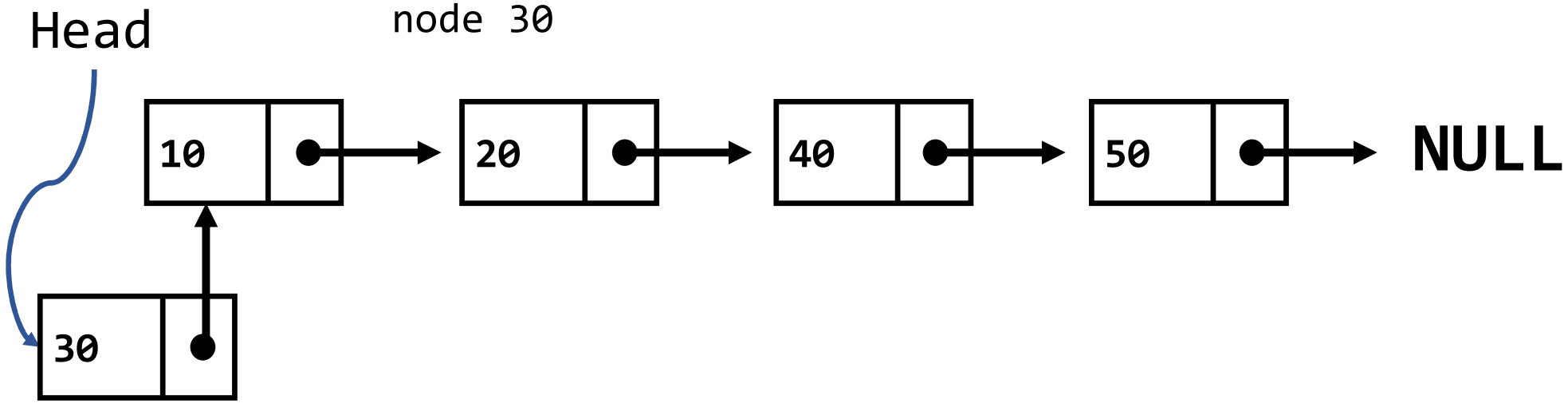
So we can simply make a link from 30 to 10 and change Head pointer to node 30



LINKED LIST

Insertion in Linked List:-

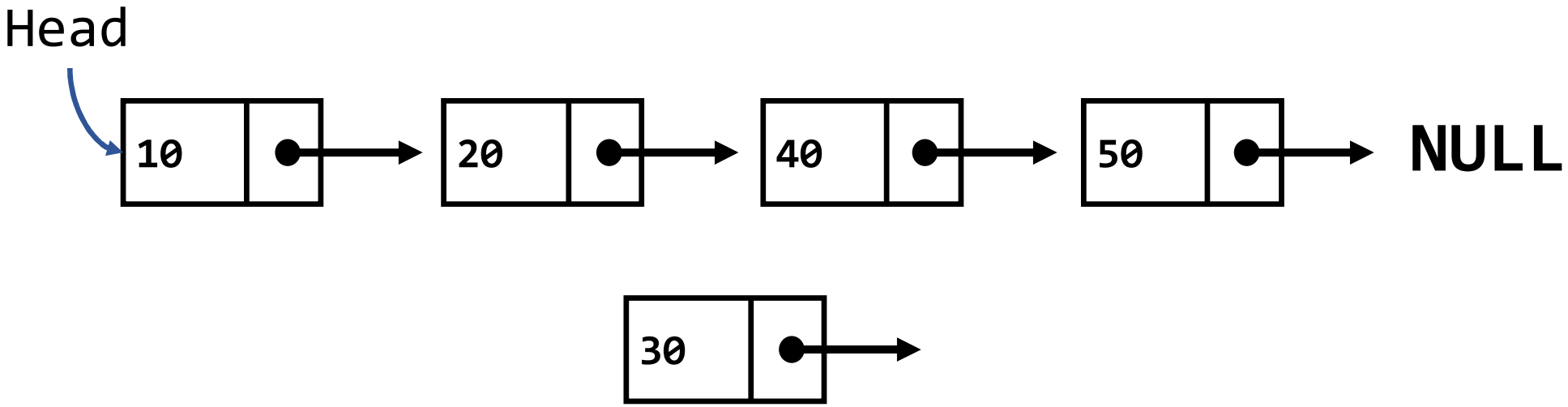
So we can simply make a link from 30 to 10 and change Head pointer to node 30



LINKED LIST

Insertion in Linked List:-

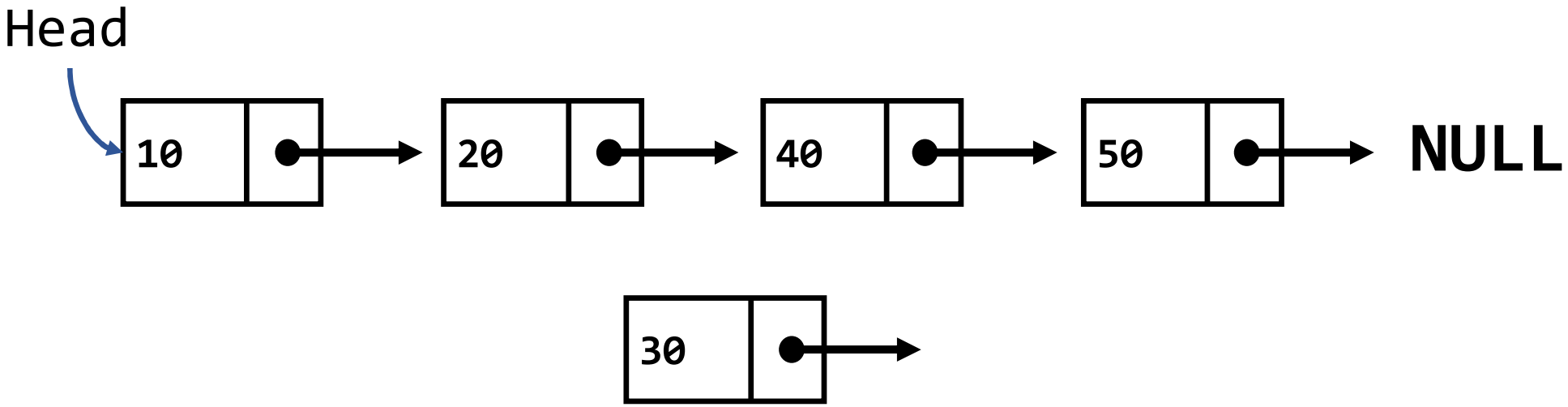
Now let’s say we want to insert a node after 20



LINKED LIST

Insertion in Linked List:-

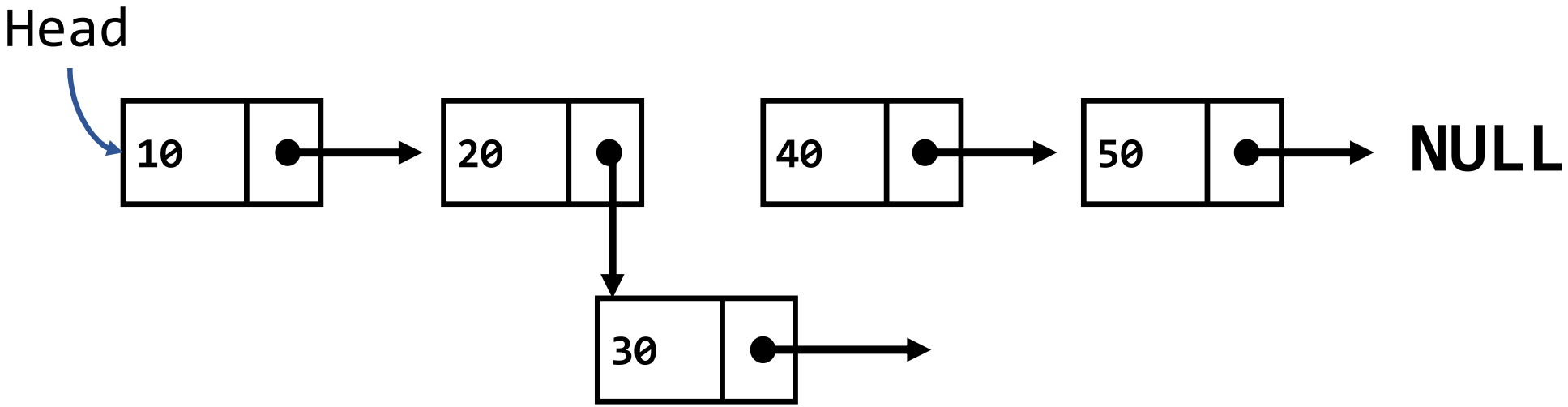
Here we can remove link between 20 and 40
and make it from 20 to 30 and then 30 to 40



LINKED LIST

Insertion in Linked List:-

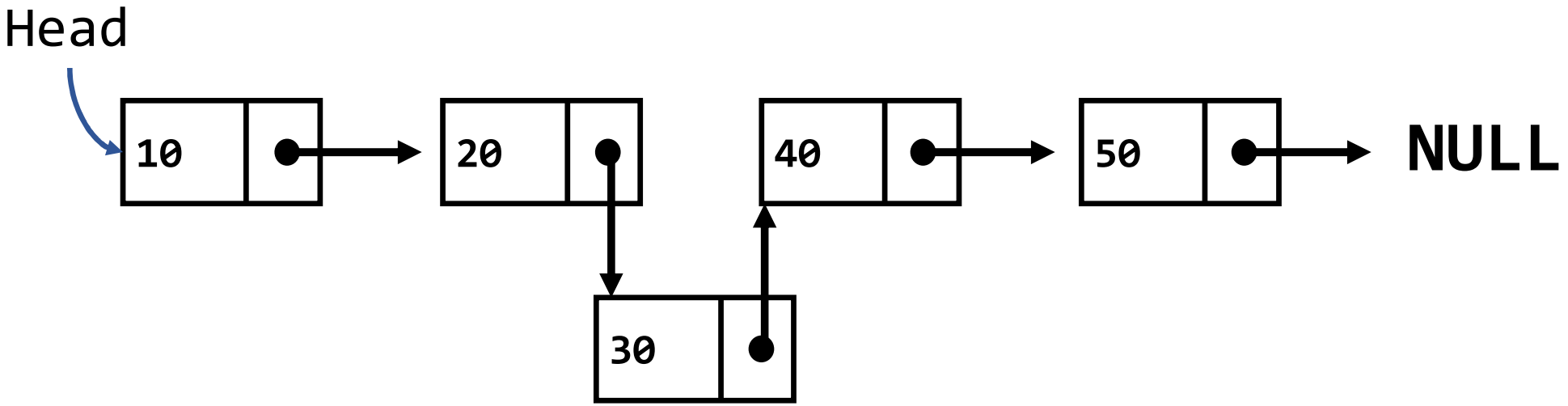
Here we can remove link between 20 and 40
and make it from 20 to 30 and then 30 to 40



LINKED LIST

Insertion in Linked List:-

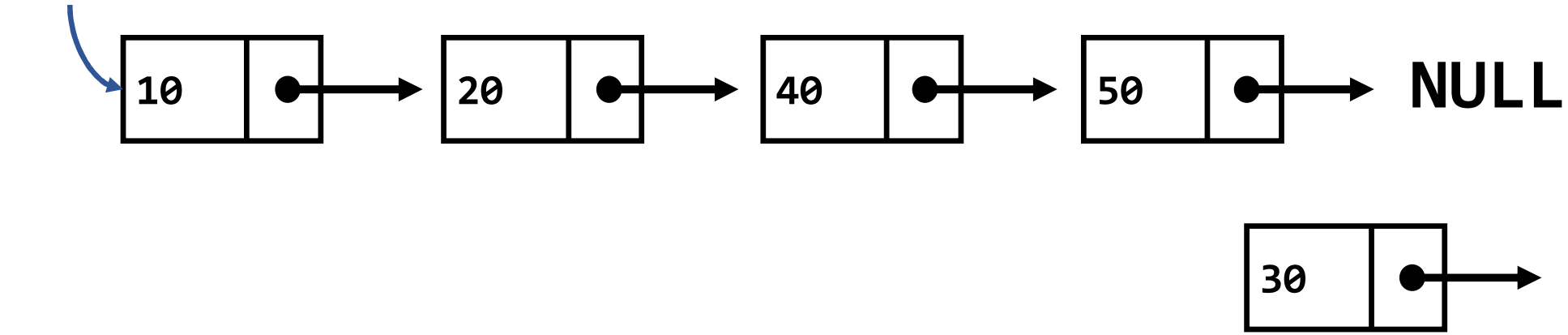
Something like this



LINKED LIST

Insertion in Linked List:-

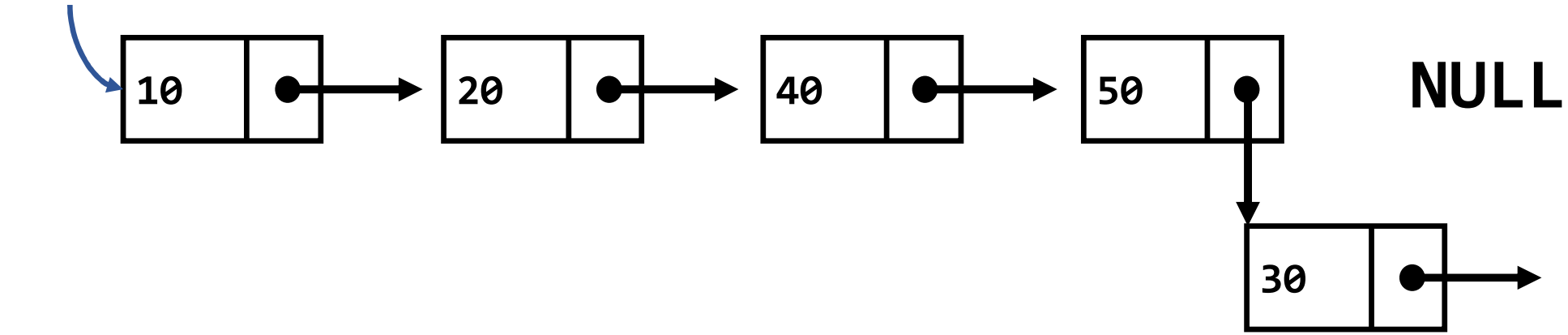
Now if we want it to insert at the end then we can simply create a link from 50 to 30 make next pointer of 30 as NULL



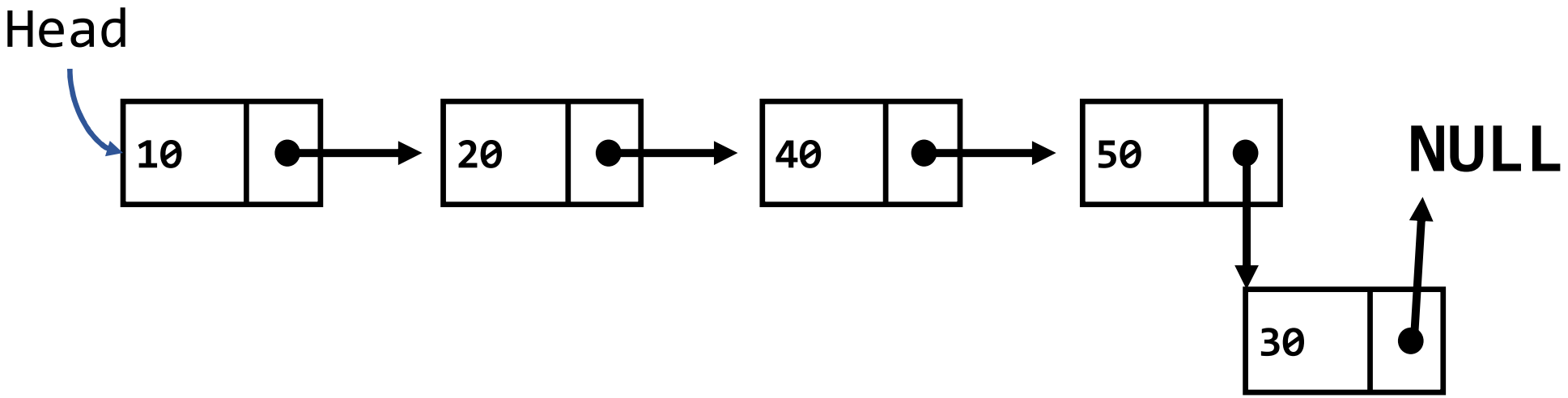
LINKED LIST

Insertion in Linked List:-

Now if we want it to insert at the end then we can simply create a link from 50 to 30 make next pointer of 30 as NULL



LINKED LIST

Insertion in Linked List:-

It will look like this