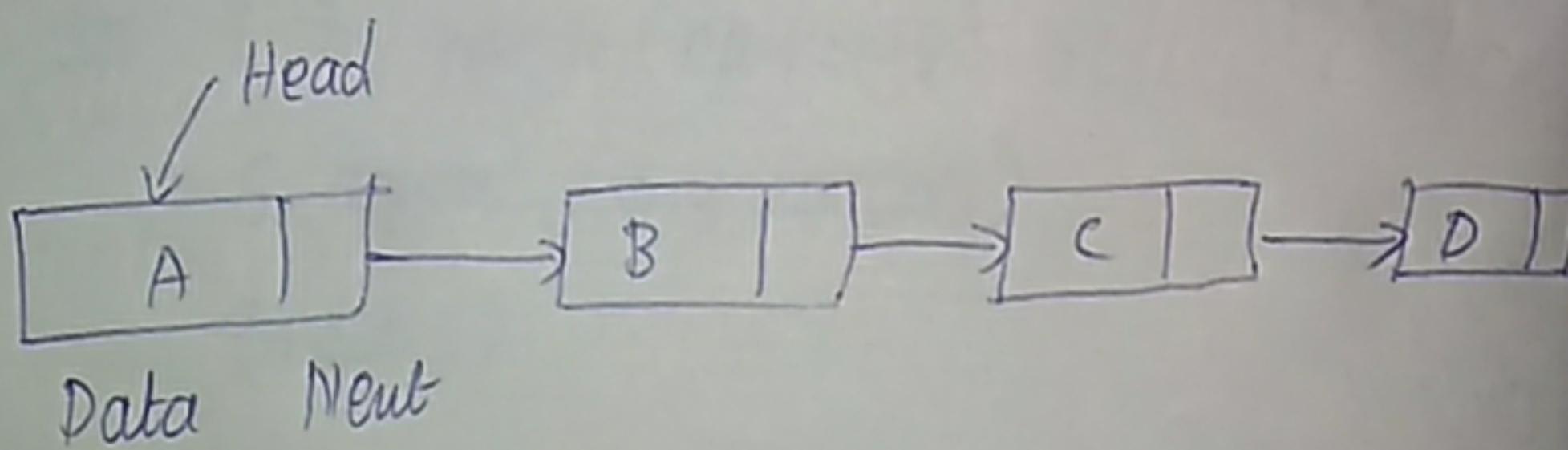


Day 10

Linked list

* A linked list is a linear data structure, in which the elements are not stored at contiguous memory location. The elements in linked list are linking pointers as shown in the below image.



* A linked list consists of nodes where each node contains a data field and a reference to the next node in the list.

Linked list vs array:

An array is a datastructures that contains a collection of data (similar type) elements whereas the linked list is considered as non-primitive data-structure contains a collection of unorder linked elements known as nodes.

Advantages:

- * Dynamic Size
- * Easy insertion / deletion

disadvantages:

- * Random access is not allowed, we have to access elements sequentially starting from the first node.
- * Extra memory space for pointers is required with each element of the list.

* Array have better cache locality than can make a pretty difference in performance.

Single linked list

- * A collection of nodes that collectively form a linear sequence.
- * Each node stores a reference to an object that is an element of the sequence, as well as a reference to the next node of the list.

Implementation:

class Node(object):

def __init__(self, value):

self.value = value

self.nextnode = None

class linkedlist:

def __init__(self):

self.head = None

def printlist(self):

temp = self.head

while (temp):

print(temp.data)

temp = temp.next

Advantages:

- * linked list have constant time insertions and deletions in any position. In comparison, array require $O(n)$ time to do the same thing.
- * linked lists can continue to expand without having to specify their size ahead of time.

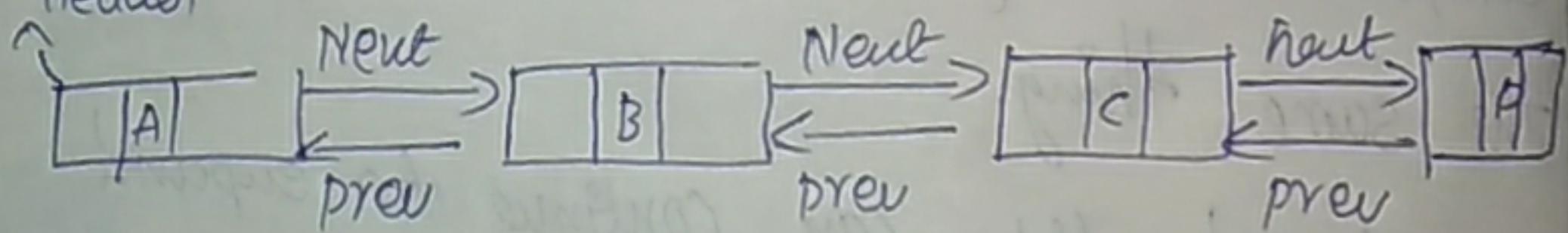
disadvantage :-

- * To access an element in a linked list, you need to take $O(k)$ time to go from the head of the list to the k th element.

Doubly linked list :

Doubly linked list contains an extra pointer, typically called previous pointer, together with next pointer and data which are the singly linked list.

Header



Implementation :-

```
class Doublylinkedlist:
```

```
    def __init__(self, value):
```

self.value = value

self.nextnode = None

self.prevnode = None

Advantages:-

* Traversed in both forward and backward.

* delete option

* quickly insert newnode before a given node.

Disadvantages:-

* Every node of DLL require extra space for an previous pointer.

* All operations require an extra pointer previous to be mentioned.