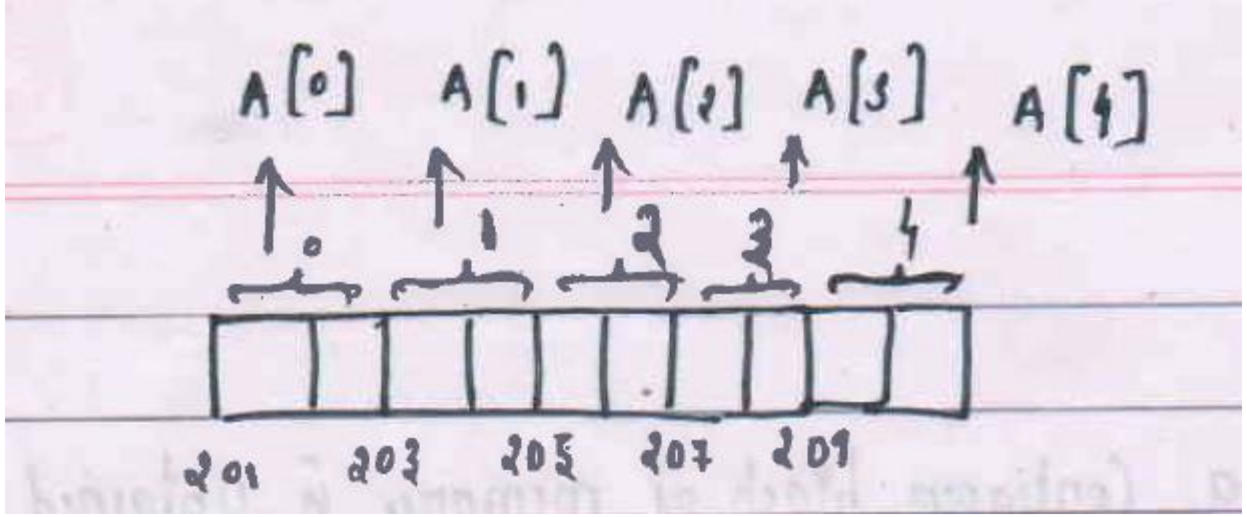
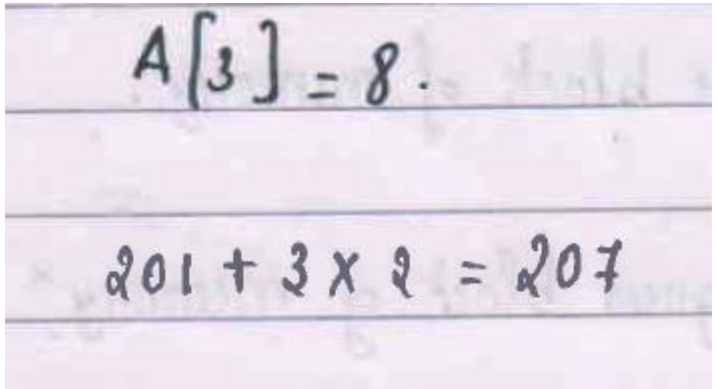
Linked List

Limitation of array

Consider the array declaration, int A[5]

When Memory manager sees this declaration, it will look for 10 bytes of contiguous memory.(suppose starting address of A is 201.)

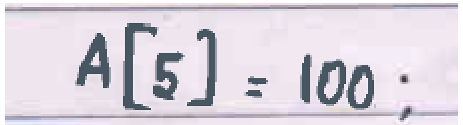




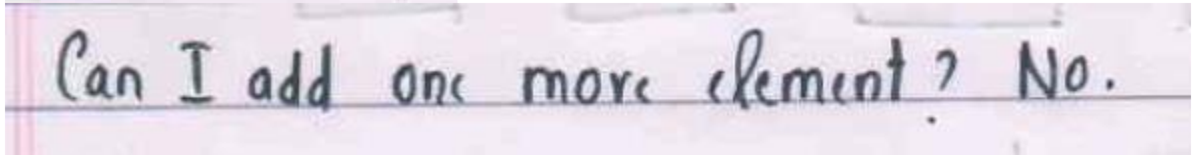
A[3] = 8.

201 + 3 x 2 = 207

Base address+ index of the element* number of bytes



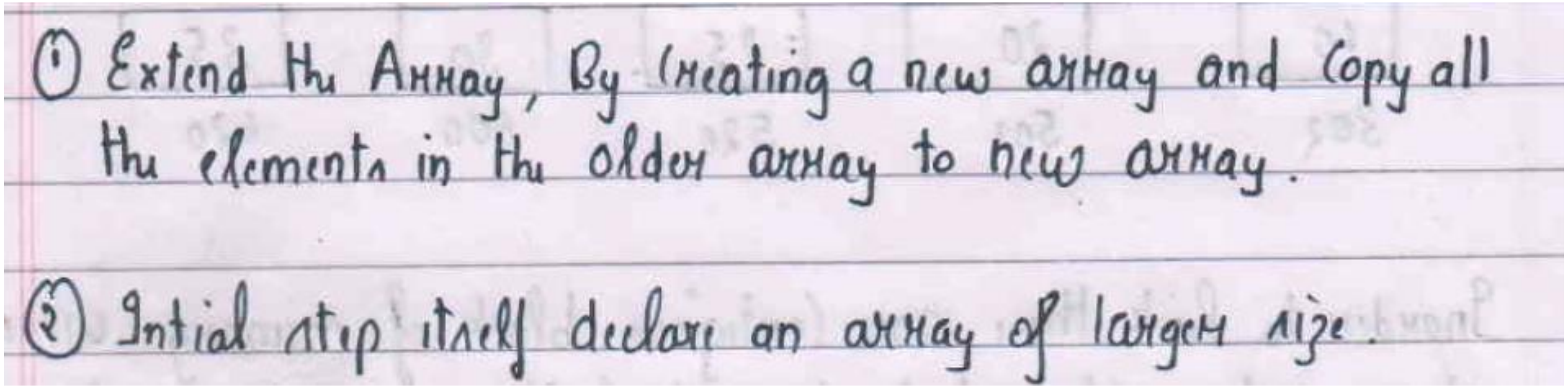
A[5] = 100 ;



Can I add one more element ? No.

Array Index out of bound error.

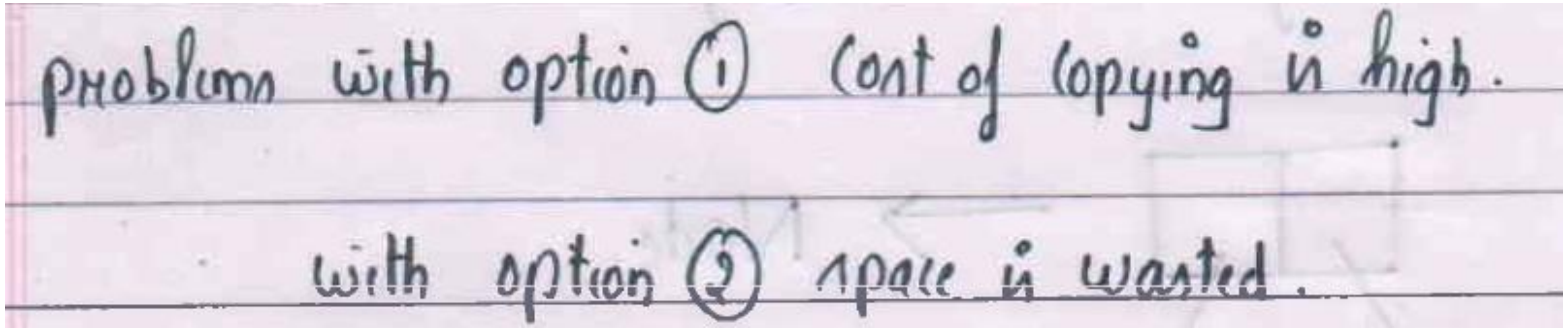
Options



① Extend the Array, By Creating a new array and Copy all the elements in the older array to new array.

② Intial step itself declare an array of larger size.

Problems



problems with option ① Cost of Copying is high.

with option ② space is wasted.

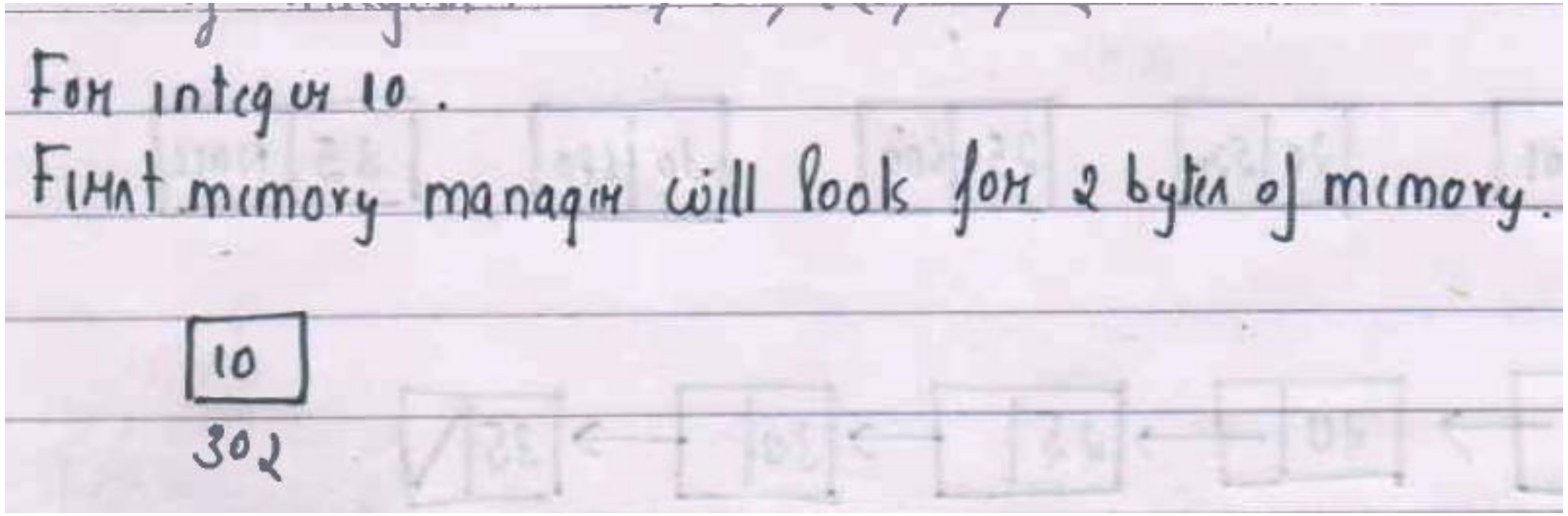
Solution to this problem: **Linked List Data Structure**

Suppose the list of integers is **10,20,25,30,35**

In **array** for the above list we will get **contiguous block** of memory.

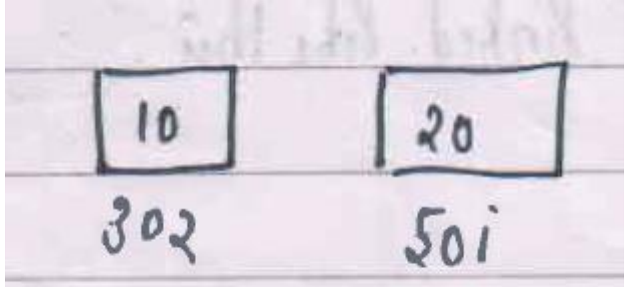
In **linked list** instead of getting one block of memory, we gets memory for **one unit of data at a time**.

For integer 10 in the list a separate memory is allocated. For integer 20 in the list a different memory unit is allocated etc

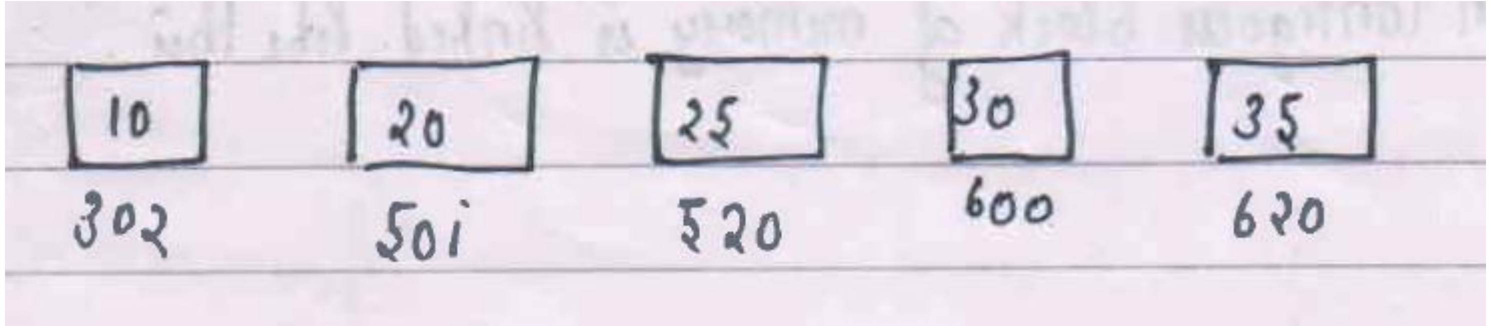


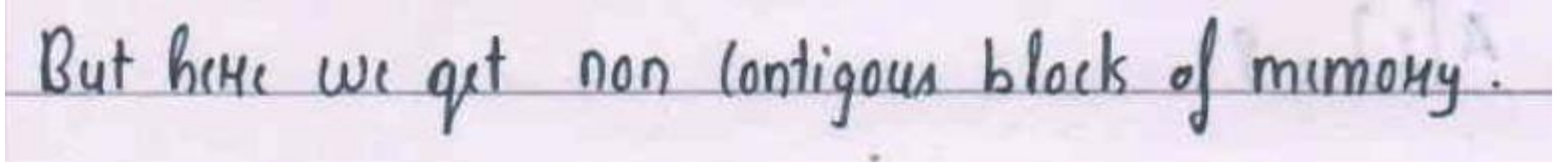
For integer 10.

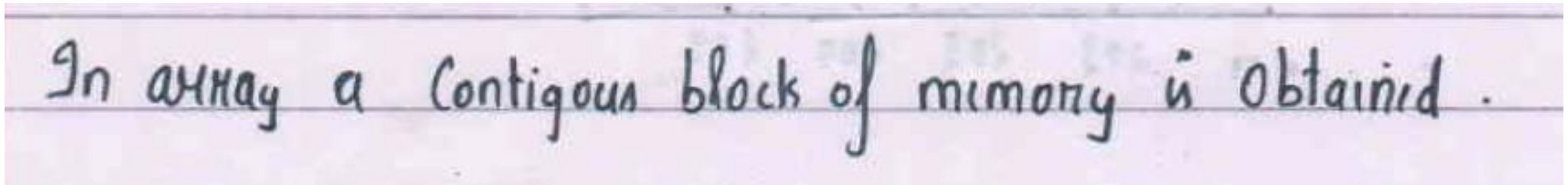
First memory manager will look for 2 bytes of memory.



10,20,25,30,35

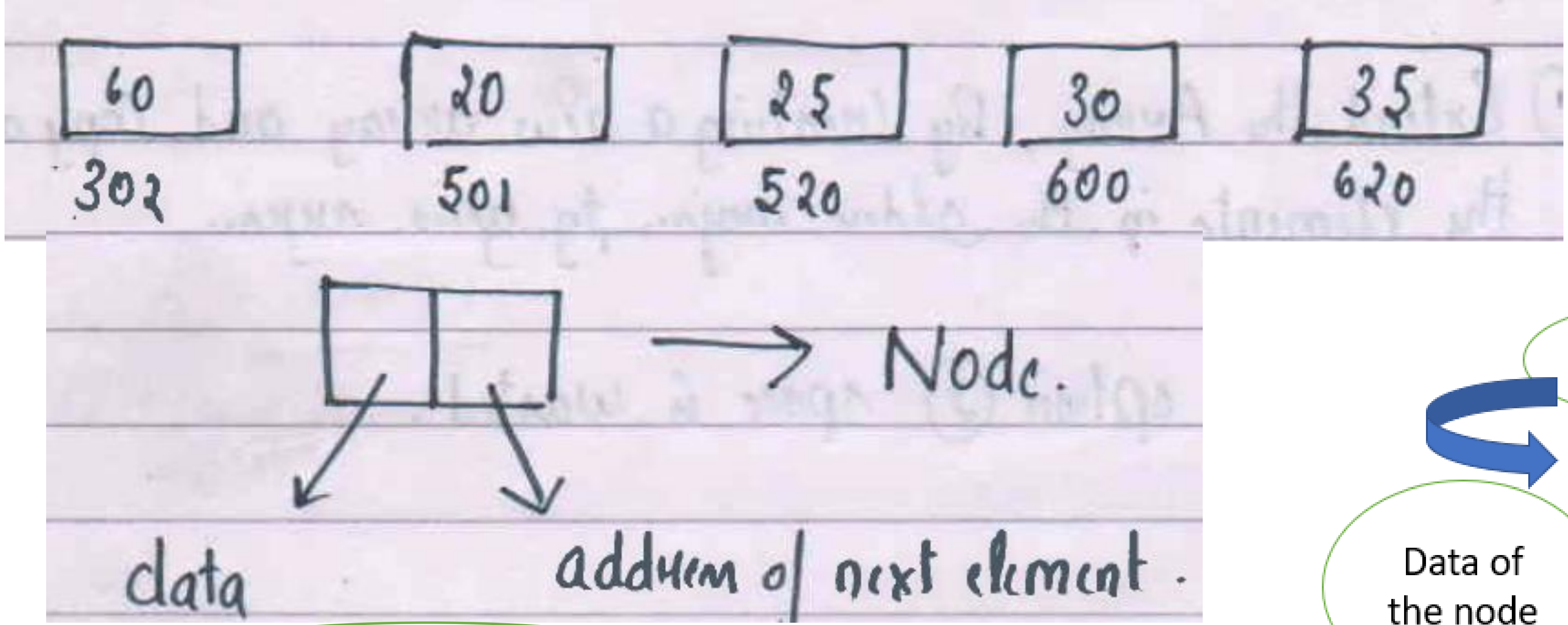



But here we get non contigous block of memory.

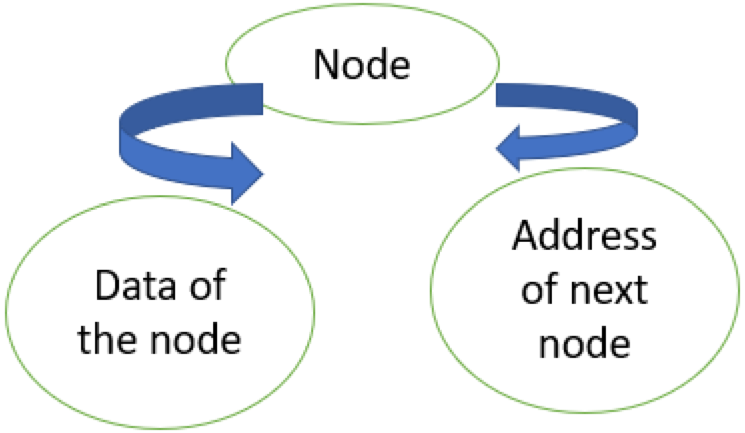

In array a contigous block of memory is obtained.

How to access this non contiguous blocks of memory?

To access this non contiguous blocks of memory we need extra field for an element in addition to the value stored.



5 nodes in the above ex:

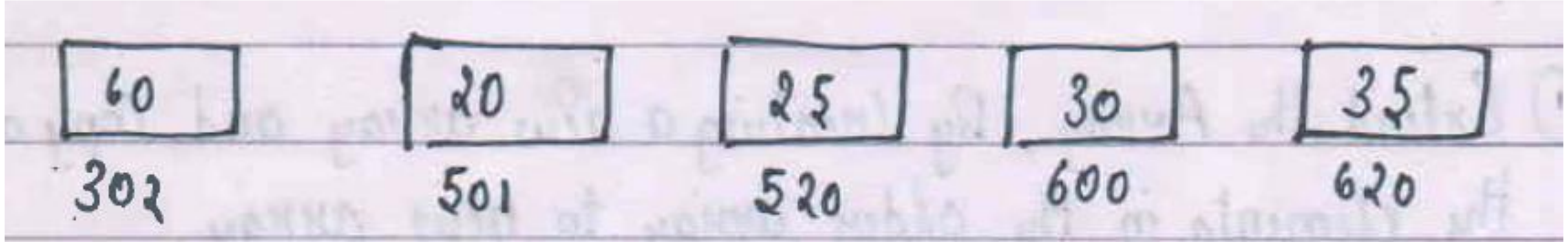


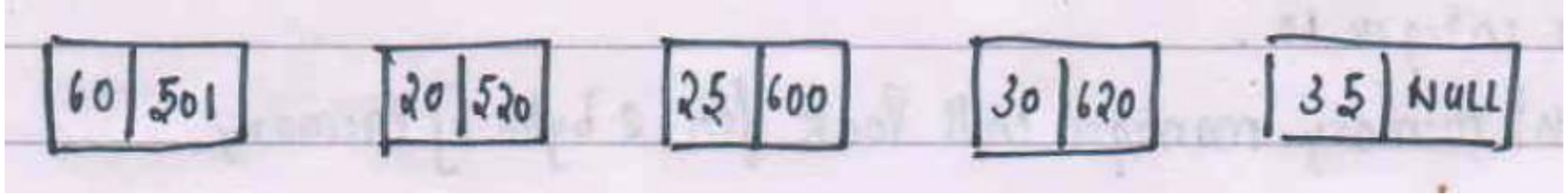
Node

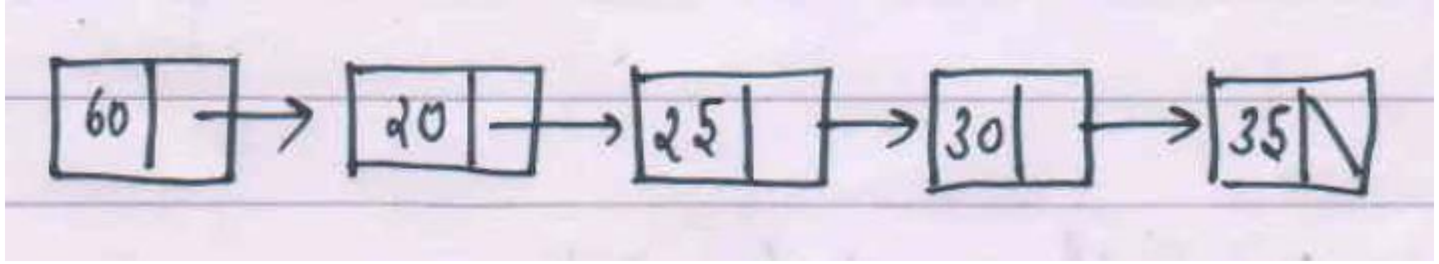
Data of the node

Address of next node

How to access this non contiguous block of memory?

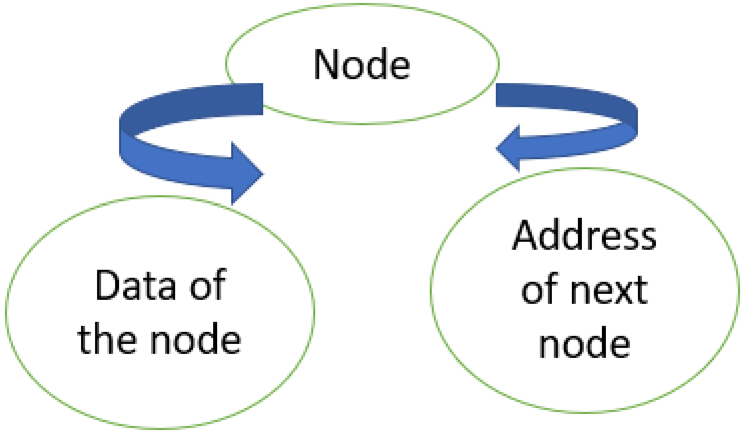






Non contiguous block of memory is linked like this

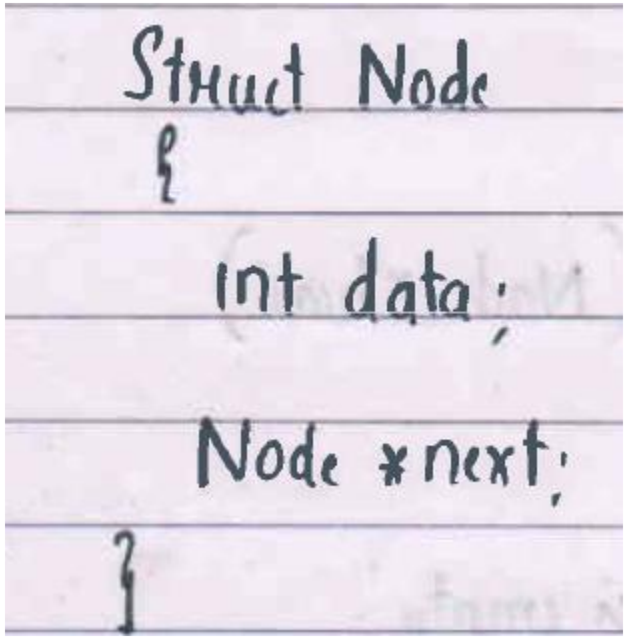
How to implement this collection of nodes?



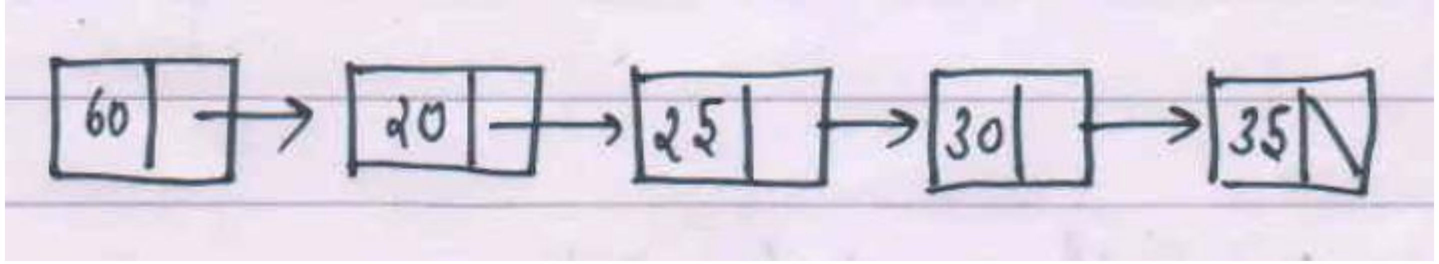
Node

Data of the node

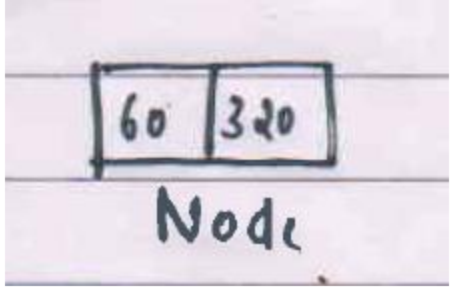
Address of next node



```
Struct Node
{
    int data;
    Node *next;
}
```

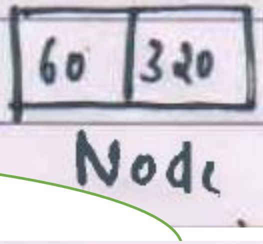


How to access a Node?



Node.data → gives the data in the node

Node.next → gives the address of the next node.

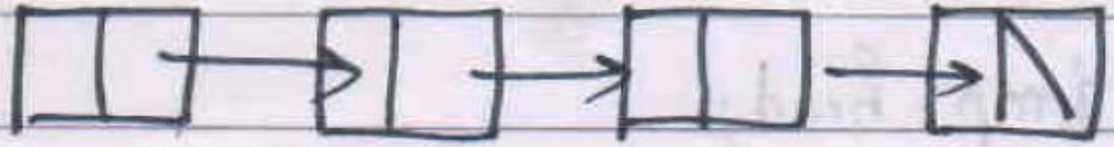


Node.data → 60

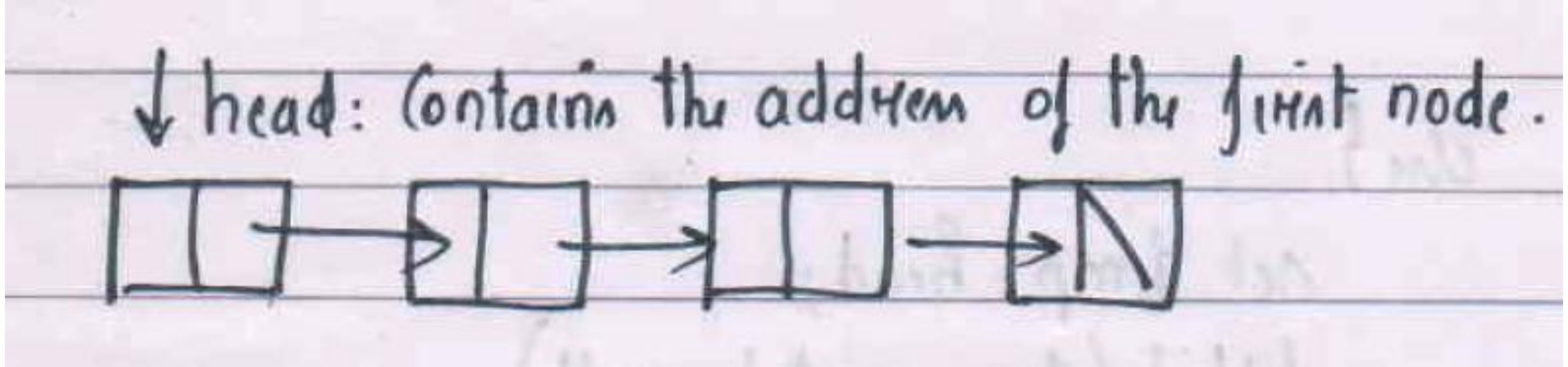
Node.next → 320

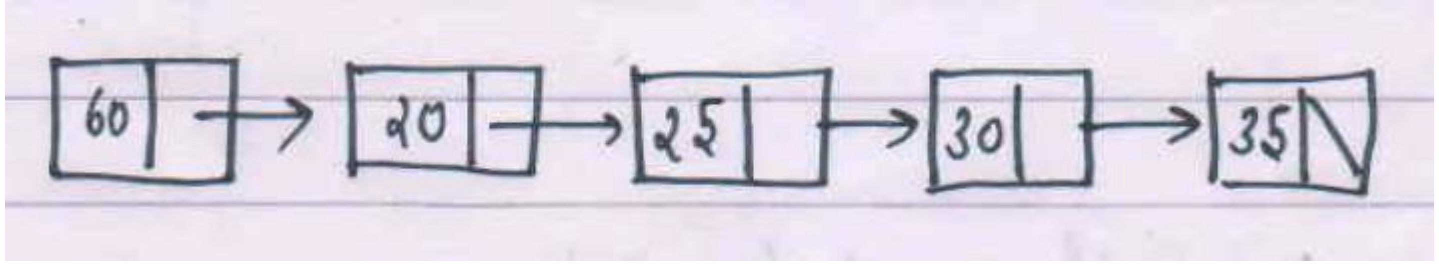
Only information we keep track of linked list is address of the first node.

↓ head: Contains the address of the first node.



How we access the List?





- Operations on linked list will discuss in the next class.

To do list

- Scribe this notes and upload it in the AUMS.
- Online exam next week. Change in portions(recursion excluded, Algorithm analysis included)