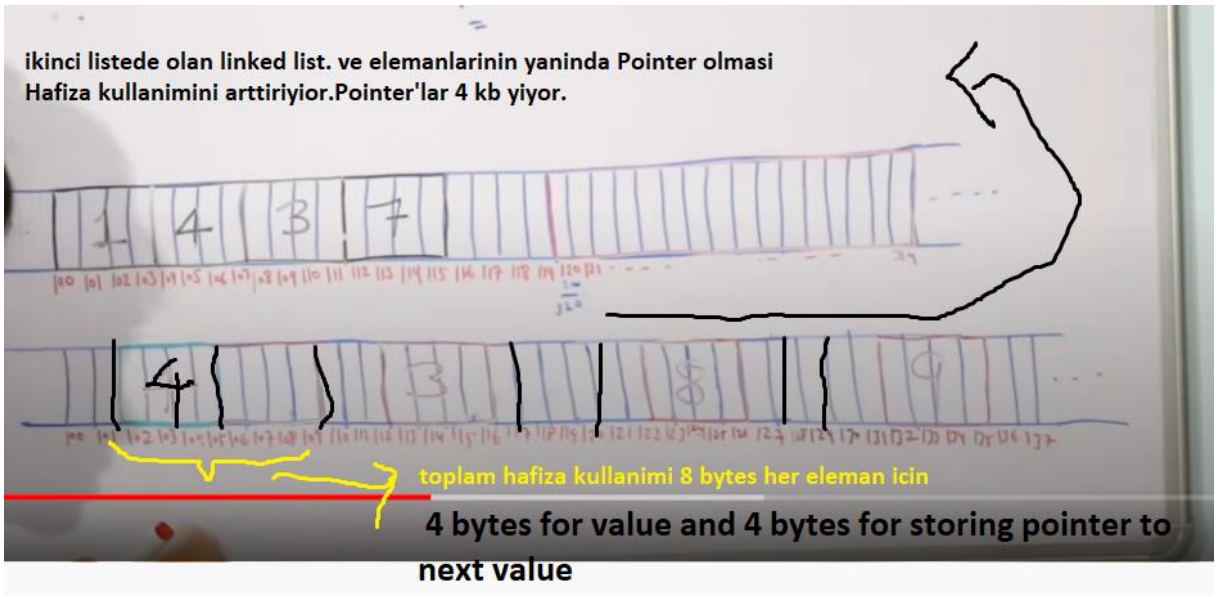
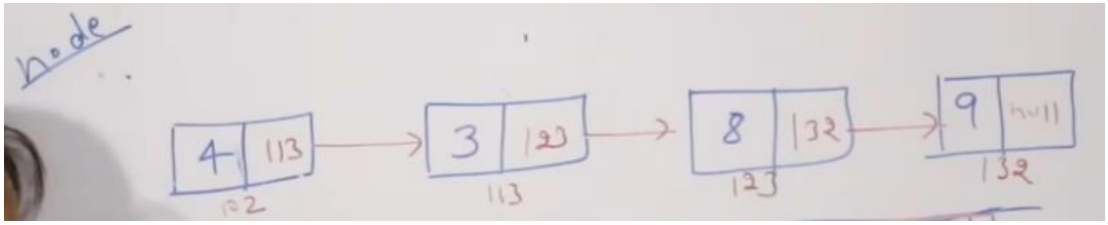
Arraylarda ornegin 50 boyutunda bir array tanimladin.Ama yalnica 10 tane index kullandin.

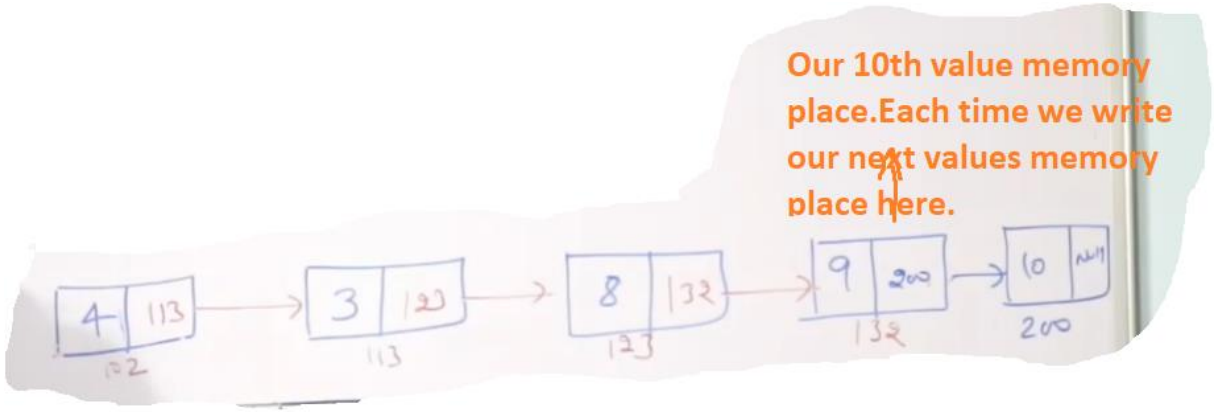
10x4=40 bytes kullanacak ama 50 x4=200 byteslik bir memory aldi ve geri kalan memory kullanilamiyor

Bu problemini cozumu linked listtir.



Simdi linked listeme 10 sayisini eklemek istedigimi varsayalim





Best case of this O(1) means you find your element on first

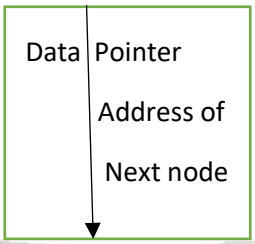
Worst case for this O(n) means you find your element on last index.

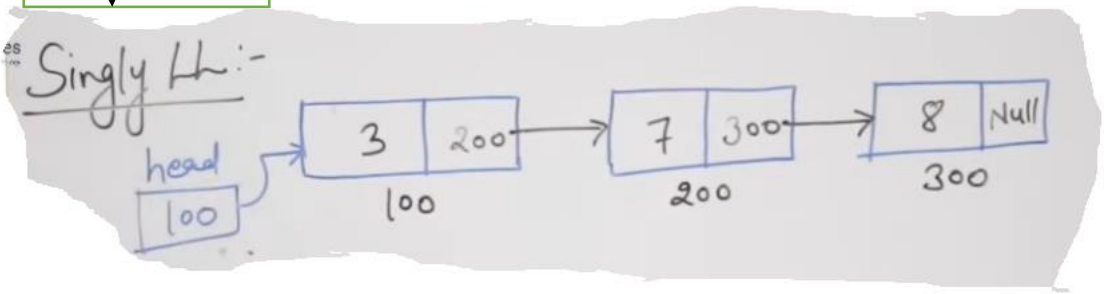
Binary search is not possible for linked lists.Because for binary search you have to find middle element.

Types of linked list in data structures

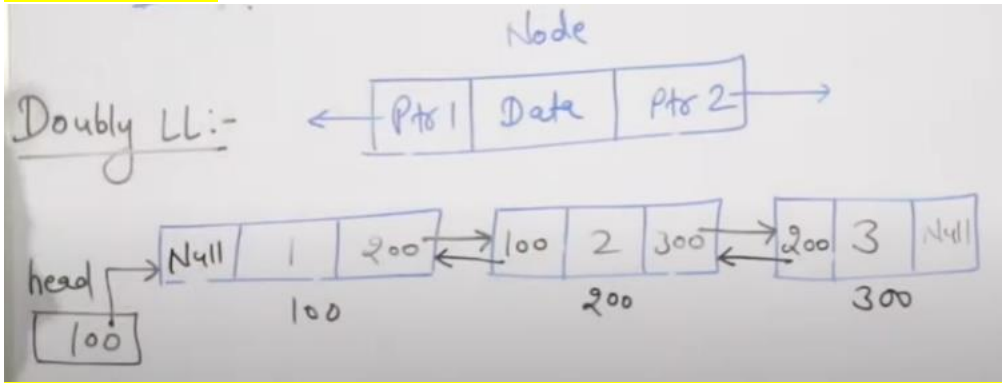
Single linked list: It contains two part,one of data and other one is address of next node.

Each node contains data part and pointer for next node.

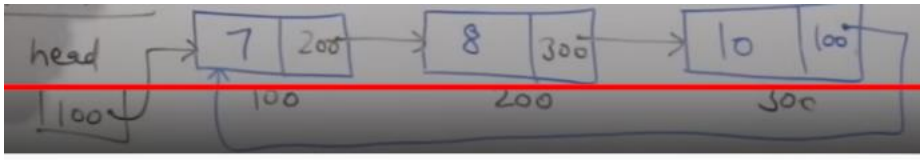




Double linked list:



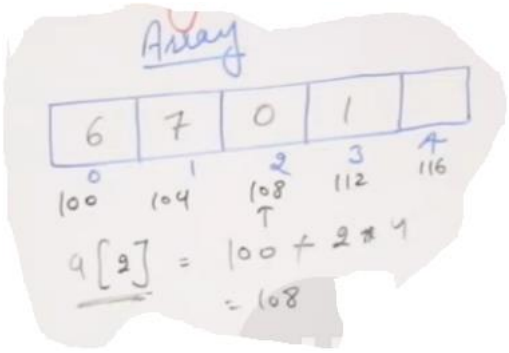
Circular linked list: it's a variation of single linked list. Difference is in single linked list you make last pointer as null but for circular linked list you make address it to first node.

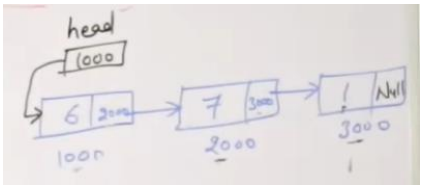


Arrays vs Linked List

We can not say Arrays is better than linked list or contraire.

Cost of accessing an element ARRAY LINKED LIST





it's constant time. O(1) For linked list lineer time O(n)

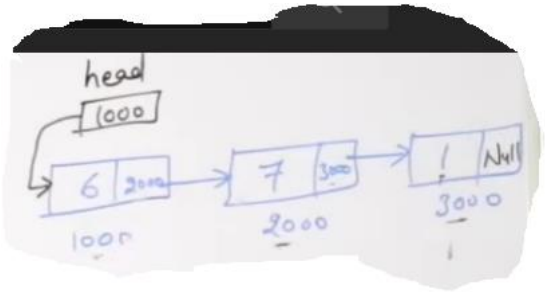
In array we see contagious block. But for linked lists no.In array data is stored in contagious block.but in lined list data is stored in ramndom positions.

2-Memory requirement,utilization

1-MEMORY UTILISATION LINKED LIST BETTER THAN ARRAY

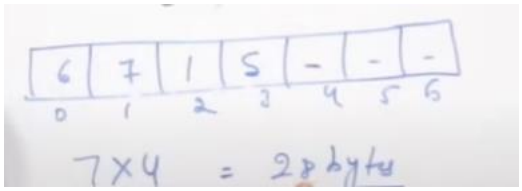
Array are fixed sizes.

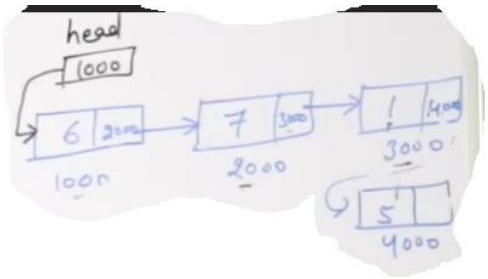
-Memory utilization in Array is worst than Linked lists because for example u create an erray 20 size but u use just a 10 size of this array.That means u reserve 20 memory from your computer and u use just 10 it means Memory wasting.



*For linked list we need space for pointer also so for example for an array with 3 elements we need 3x8=24 but if u have a 3 sized array and u use them all then 3x4=12

Ex:We want to add 5 to array and linked list.





If u add 5 to array u don't need to increase array because you have already had fixed size array and you have places to elements so It stays on 28 bytes

For linked list,If u want to add 5 to array you have already 24 bytes 3x8 4 for data 4 for pointers.Then you add 5 and you use 32 Bytes.

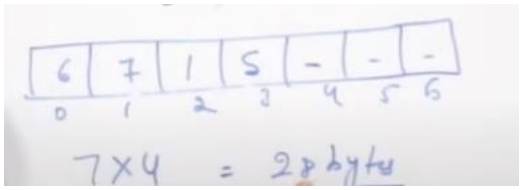
If you u use 7 elements of array it means 7x4=28 bytes

If you use 7 elemtens on linked list it means 7x8=56 bytes .

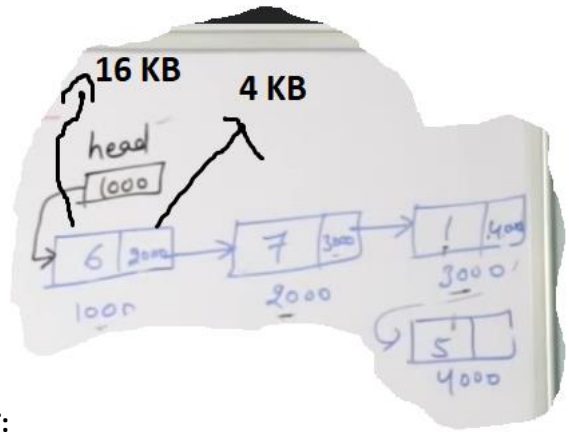
In this case memory requirement is less than linked lists.

But memory utilization is efficient for linked lists than arrays.

3-COMPLEX DATA TYPES; it takes 16 bytes to store.

ARRAY : 

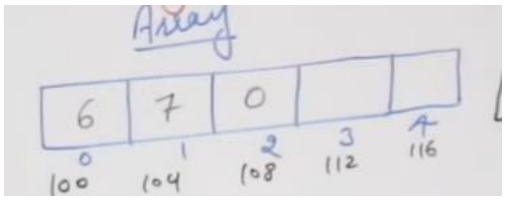
16x7= 112 bytes



LINKED LIST: **4x20=80 KB**

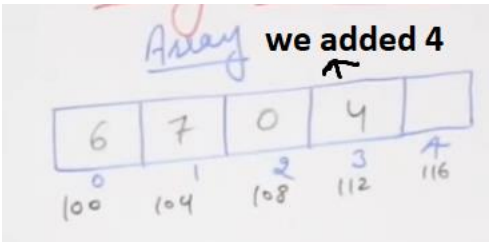
The size of data insert large it is better to use linked list. Linked list consume less memory than Array.

4-COST OF DELETION AND INSERTION



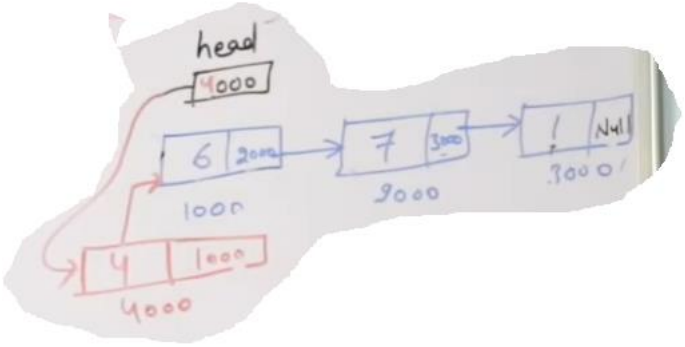
Array

A-at beginning: you have to shift all of elements So Time complexity O (n)

B-At end= O(1) 

C-At I th position : middle also is O(n)

Linked list



A-At beginning: O(1)

B -at end=You have to traverse all of list so Time complexity O(n)

C- at I th position: O(n)

FOR DELETION FOR ARRAYS O(n) but for deletion for Linked list O(1) BEGINNING

For deletion At the end ARRAY O(n) Linked list O(n)

For deletion middle for array O (n) for linked list O(n)

WHICH IS EASY TO USE ARRAYS OR LINKED LISTS ?

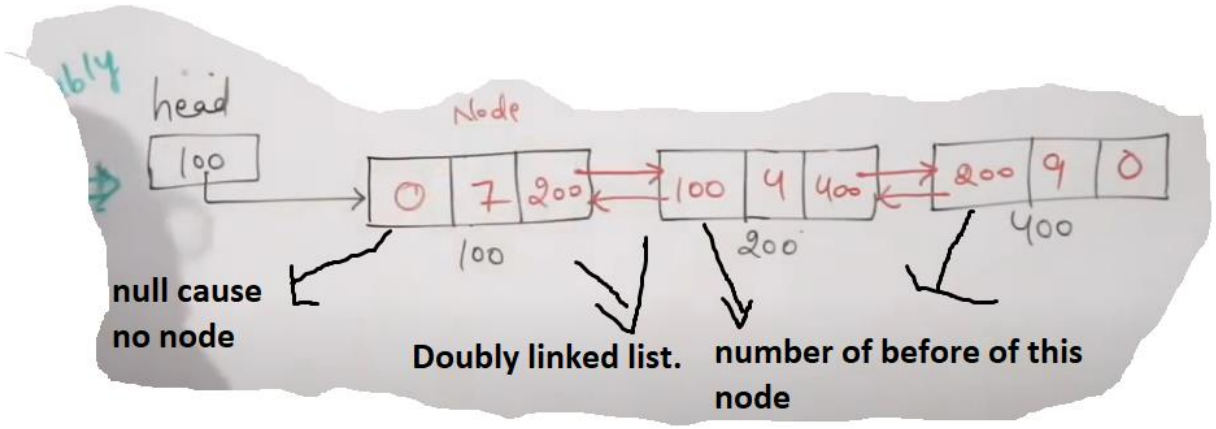
Arrays are easy to use than linked lists.

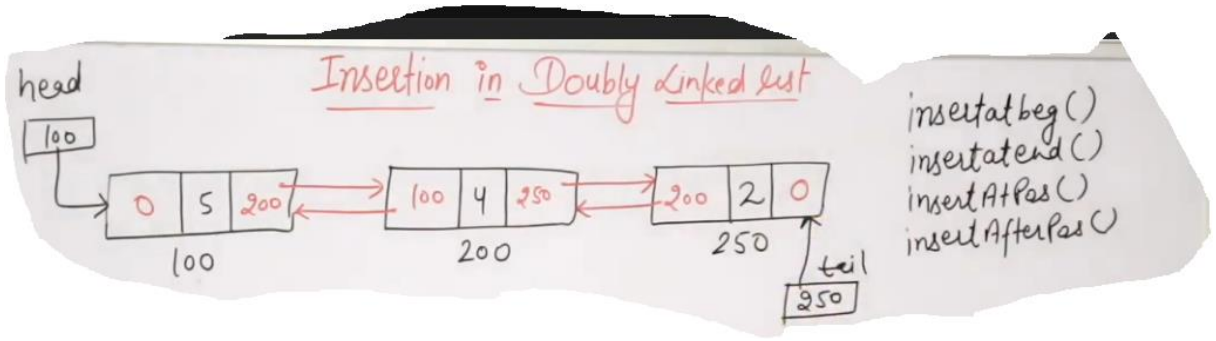
Searching operations

Array; binary and lineer search are possible

Linked list: Only lineer search Is possible**.**

2.9 Introduction to Doubly Linked List - Data structures





MAINTANING TAIL POINTER IS IMPORTANT THING

If you maintain to tail then Insertion to end takes O(1) constant time

If you do not maintain to tail then insertion to end takes O(n)

If you want to delete last node and If you maintain tail time complexity O(1)

If you want to delete last node and If u do not maintain tail time complexity O(n)

DELETION FROM DOUBLED LINKED LIST