LINKED LISTS

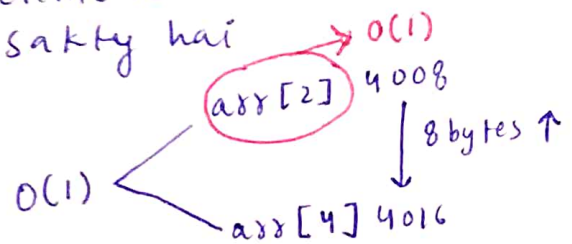
Array

```
int [] arr = new int [10];
```

Array heap memory me contiguously (continously) saare ek sath elements ko store krta hai!

Array ka hamesha FIXED size rahega! Hum array ko chota ya bada nahi kr sakty!

O(1) me hum koi bhi element ko select kr sakty hai



∴ O(1) + K times
= O(1) times

add of arr[4] = add of arr[2] + 8 bytes (K times)

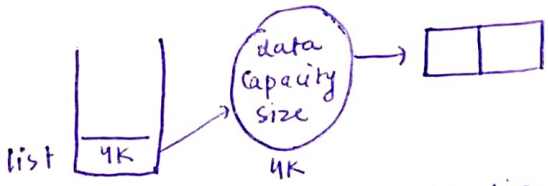
Time O(1) hi lagega arr[2] par jaane me!

ArrayList

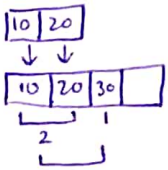
* Arraylist me size Dynamic hota hai
* Heap Memory me continously store hogi Arraylist bhi!
* Arraylist me size grow kr sakta hai dynamically.
* Capacity aur size are 2 different things in Arraylist.
  - Total no of cells
  - The no. of cells that have data in them

```
ArrayList<> list = new ArrayList<>();
```

Let initial Capacity = 2
Let increment in Capacity = 2 times capacity if size == capacity



→ list.add(10) → 1 operation | 10 |
→ list.add(20) → 1 operation | 10 | 20 |
→ list.add(30) → ∴ Now double the capacity



3 operations

→ list.add(40) → 1 operation | 10 | 20 | 30 | 40 |

→ ∴ 2 add = 4 operations

LINKED-LIST

- it is of Dynamic Size.
- it stores the elements in non-continous way in heap memory.

it can utilize the Fragmented memory.

Jab memory me SPACE tukdo me available ho usse bolty hai Fragmented memory

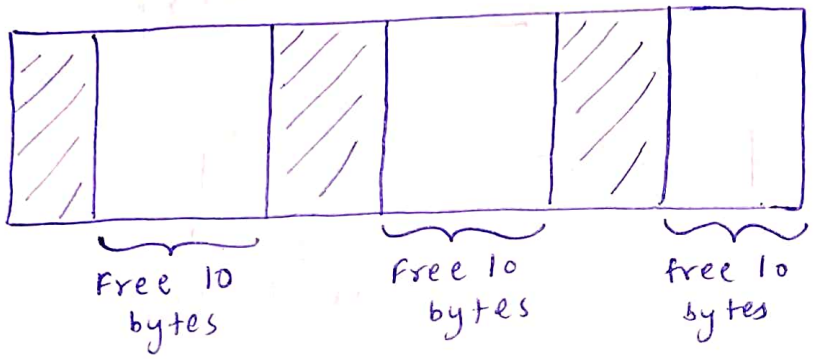
∴ Aur Fragmented memory me Array and Arraylist cannot store bcoz the space is not enough!

Whereas, LINKED-LIST can store all it's elements in fragmented memory!

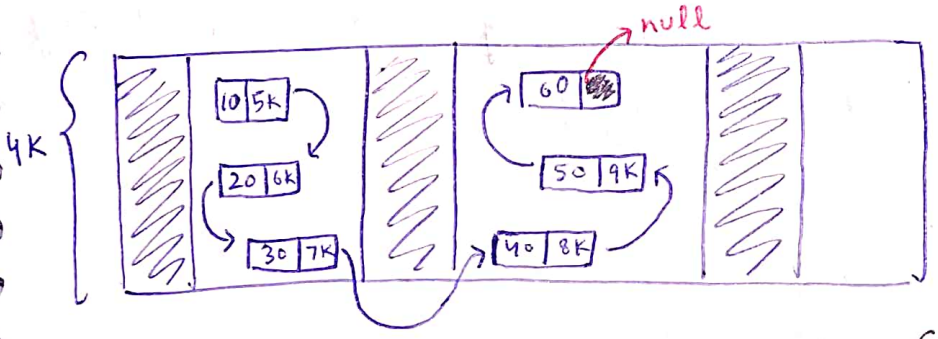
we have Array of 6 size.
↳ 6 elements of int type!

∴ 6 X 4 bytes = 24 bytes

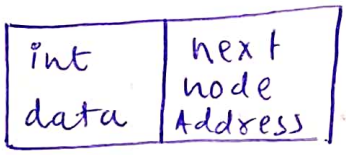
Continous 24 bytes ka space memory me kabhi available nahi hoga!
Heap memory will be as fragmented memory!



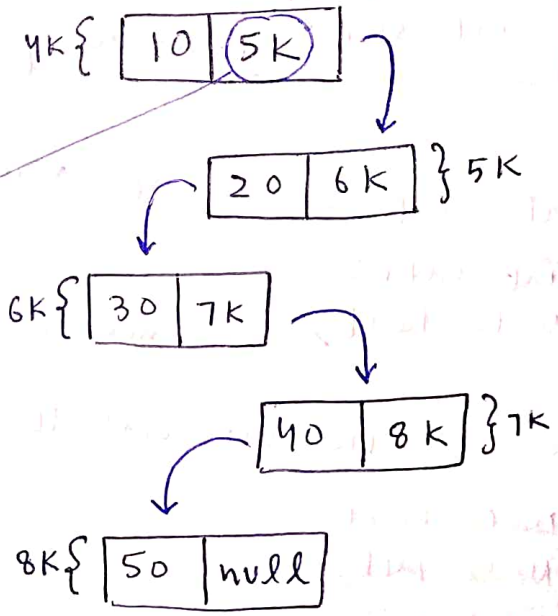
→ This is called Fragmented Heap Memory.



∴ Each node contains the element (value) and address of next element.



NODE



Demerit

- next node ka address store krne ke liye extra memory use hui hai! Jisse memory waste hui hai more than array and arraylist.

Merit

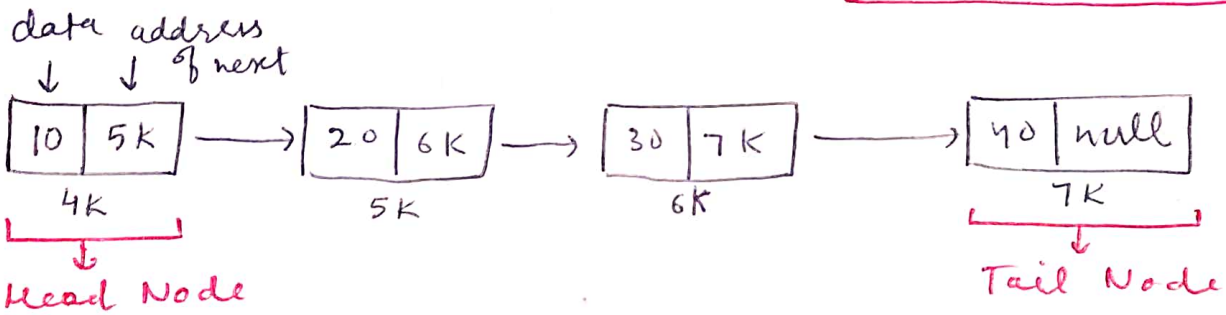
- Linked List uses the fragmented memory very optimally which was considered as waste for array and arraylist.
- Linked List also used the Dynamic Memory Allocation.

LINKED LIST KESE USE KRENGY ?

- Hum 2 class Banayegy! < Node class, Summary class

- Node class consists :
  - data
  - address of next node

```
Node
    int data;
    Node next;
```



- Head Node, Tail node, size of Node ko Maintain krney ke liye Summary class Banayengy !

```
LinkedList
    Node Head;   → address of Head node
    Node Tail;   → address of Tail Node
    int size;    → size of the node
```

```
private static class Node
{   int data;
    Node next;
}

private static class LinkedList
{   Node head;
    Node tail;
    int size;
}
// function to print a linkedlist
void display()
{   for (Node temp = head; temp != Null; temp = temp.next)
    {   System.out.print(temp.data + " ");
    }
  System.out.println();
}
```