

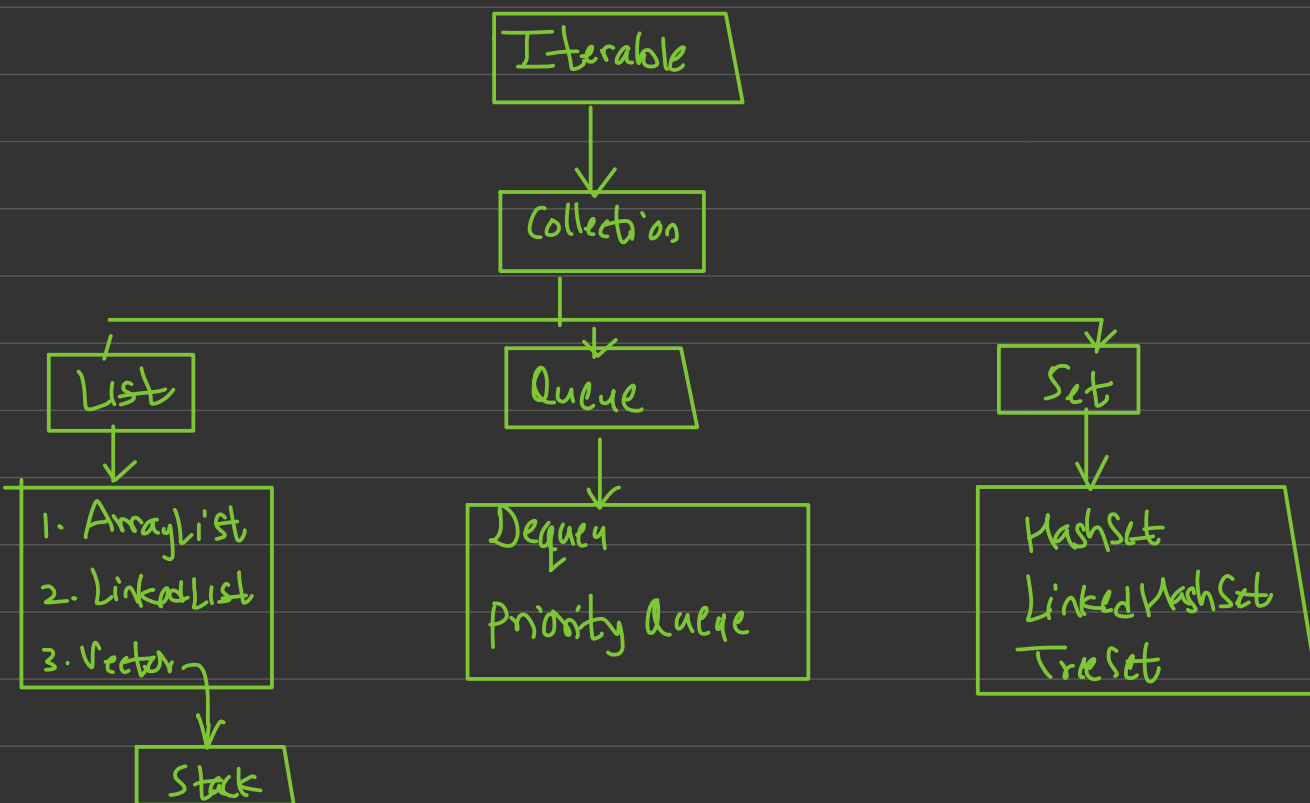
## Collection Class

Framework -> store and manipulate group of objects

> searching, sorting, insertion, manipulation and deletion..

Java Collection framework: Set, List, Queue, Deque, ArrayList, Vector, LinkedList, PriorityQueue, HashSet, Trees.

Collection: represents a single unit of objects i.e. group



To instantiate List interface, we do!

```
List<int> l1 = new ArrayList();
```

```
List<int> l2 = new LinkedList();
```

```
List<int> l3 = new Vector();
```

```
List<int> l4 = new Stack();
```

## Linked List:

Array:

	0	1	2	3	4
q	10	20	30	40	50
	112	116	120	124	128

Insert 15 b/w 10 & 20 i.e. insert at index 1

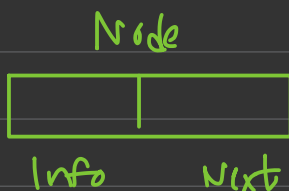
	0	1	2	3	4	5
q	10	15	20	30	40	50

for (i = 4; i >= 1; i++)  
a[i+1] = a[i]

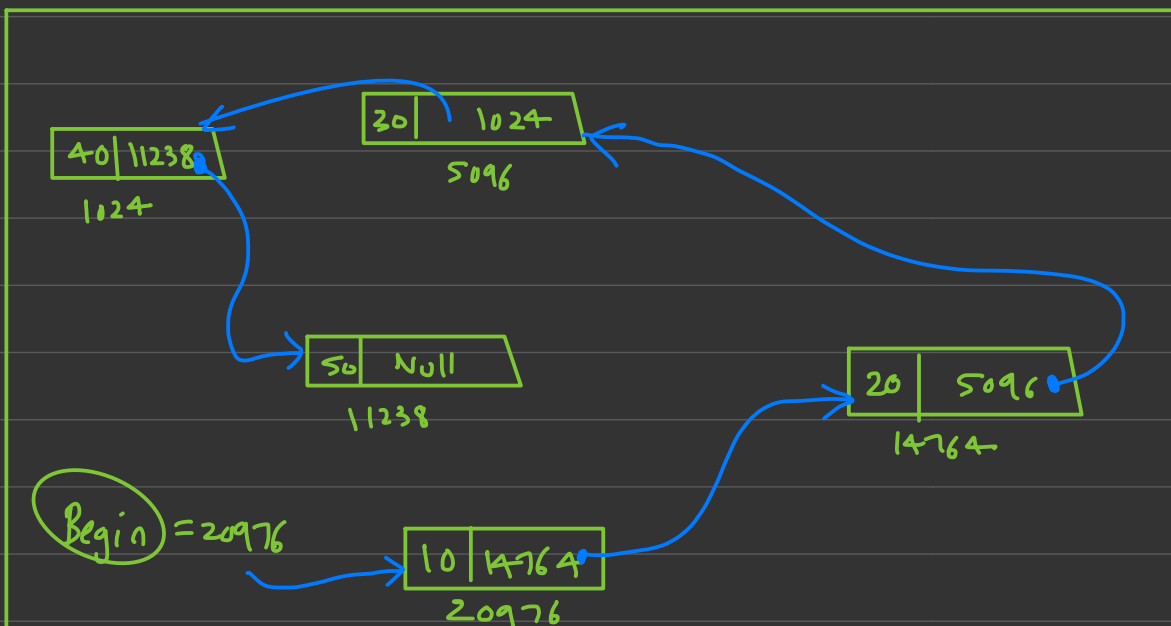
Delete 20 i.e. element at index 2

for (i = 2; i < a.length; i++)  
a[i] = a[i+1]

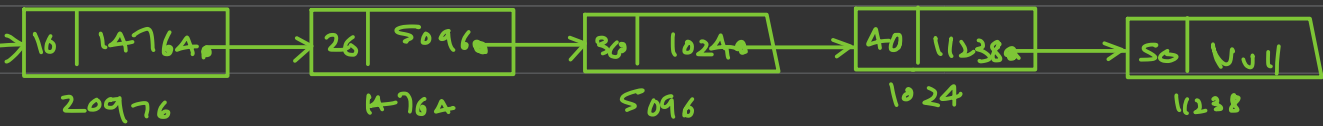
## Linked List



RAM



Begin



10 June 2023

Traverse a LL:

1. If  $\text{Begin} = \text{Null}$   
    print 'LL is empty'

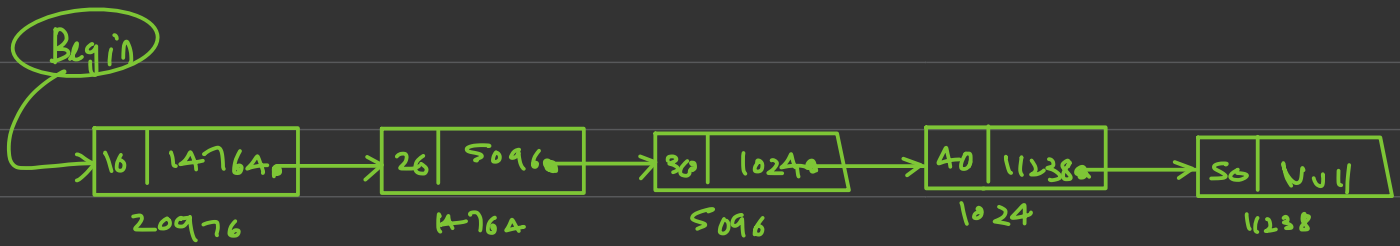
2. Set  $p = \text{Begin}$

3. while( $p \neq \text{null}$ )  
    {  
        print ( $p \cdot \text{info}$ )  
         $p = p \cdot \text{next}$   
    }

4. Stop

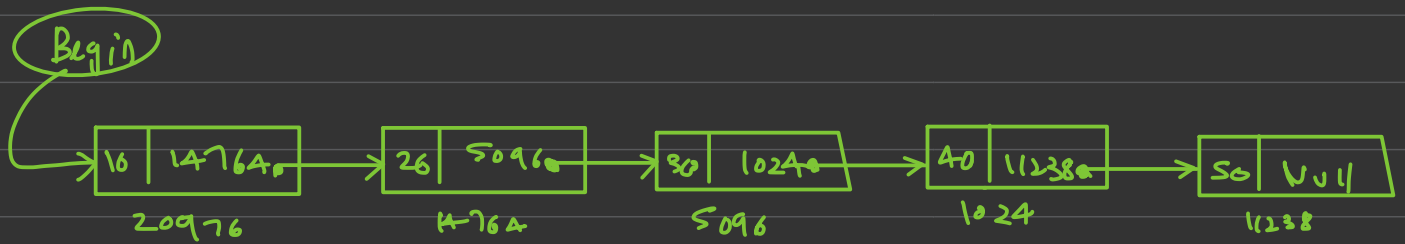


## Searching:



1. What to search: data = 40
2. If Begin = null  
    Sup "LL empty"  
    exit
3. Set P = Begin
4. Set flag = 0 // Not found
5. while (p != null)  
    {  
        if (p->info == data)  
        {  
            flag = 1; // Found  
            break;  
        }  
        else  
            p = p->next;  
    }
6. If flag == 1  
    Sup("Found at address", p);  
else  
    Sup("Not found in LL");

## Inserting a node: (end)



1. Create node 

info	
99	

  
new

2. Input new.info

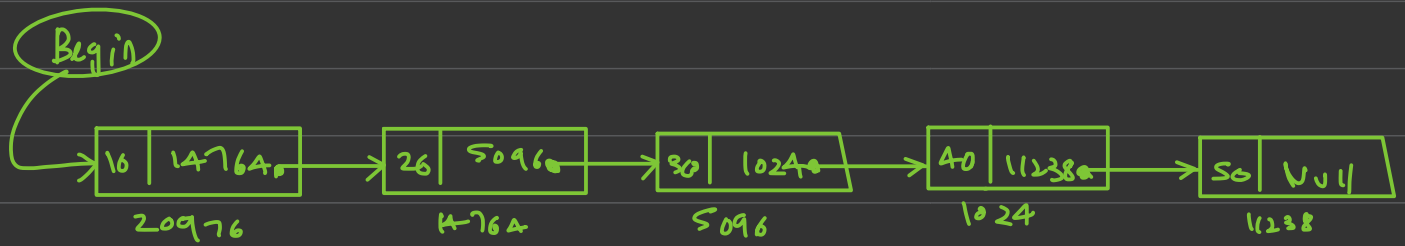
3. If  $\text{Begin} = \text{Null}$   
    {  $\text{Begin} = \text{new}$   
       $\text{new.next} = \text{null}$  //  $\text{Begin.next} = \text{null}$   
    }

4. Set  $p = \text{Begin}$

5. while(  $p.\text{next} \neq \text{null}$  )  
     $p = p.\text{next}$

6.  $p.\text{next} = \text{new}$  ;  
    $\text{new.next} = \text{null}$  ; //  $p.\text{next.next} = \text{null}$  ;

## Deleting a node:



1. Node to be deleted: data = 30

2. If  $\text{Begin} = \text{null}$   
Sop LL empty.  
exit

3. Set  $p = \text{Begin}$   
follow = null

4. while ( $p \neq \text{null}$ )  
    {  
        if ( $p.\text{info} == \text{data}$ )  
            break  
        else  
            { follow = p;  
               $p = p.\text{next};$   
            }    }

5. If  $p == \text{null}$

Sop "Data not found in LL.  $\therefore$  cannot delete"

else if ( $p == \text{Begin}$ )

