

Array's Disadvantages

Memory allocated to an array is fixed or in other words size of an array is fixed. In case of dynamic array (list), if array is full, to insert a new element, a new array should be created.

Insertion at the beginning of the array is costly ($O(n)$).

$O(n)$: in proportion to the number of elements, that amount of work is required

original array =	0	1	2	3	4	5	6
	4	6	8	0	2	12	14

Insert 3 at the beginning of the array.

0	1	2	3	4	5	6		
4	6	8	0	2	12	14		
	3	4	6	8	0	2	12	14

It can be observed that all elements are shifted by one, Now if there are millions of elements we need to shift all of them and it costs us a lot.

Deletion and Insertion from the middle is costly.

Delete 0 from middle.

0	1	2	3	4	5	6
4	6	8	0	2	12	14
	4	6	8	2	12	14

Example: Replace 'a' with 'ccc' in a given array.

Let's consider our elements to be 'adegafc'

0	1	2	3	4	5	6
a	d	e	g	a	f	a

Iteration-I:

0	1	2	3	4	5	6	7	8
c	c	c	d	e	g	a	f	a

Iteration-II:

0	1	2	3	4	5	6	7	8	9	10	11
c	c	c	d	e	g	c	c	c	f	a	

Iteration-III:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
c	c	c	d	e	g	c	c	c	f	c	c	c		

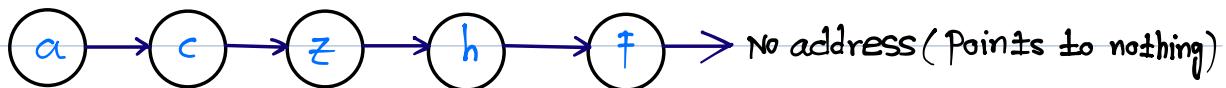
It can be observed, every time an 'a' is encountered a new array shall be created which is not memory efficient at all.

Important: When we create an array of items, we are saving addresses of those items in a contiguous locations, not the actual items or its values, and those items might be located at different locations in the main memory of computers.

Linked List

An ordered linear data structure that every item has a reference or the address of its next item. Here the addresses are not necessarily to be in contiguous location, which means items are connected like a chain.

To reach to the last in the linked, all elements that comes before it should be traversed.



How these items are connected?

a has the address of c

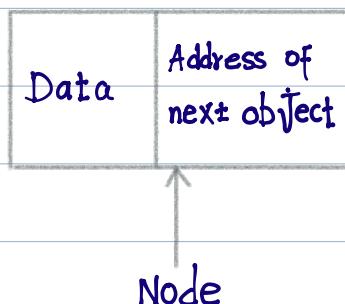
c has the address of z

z has the address of h

h has the address of f

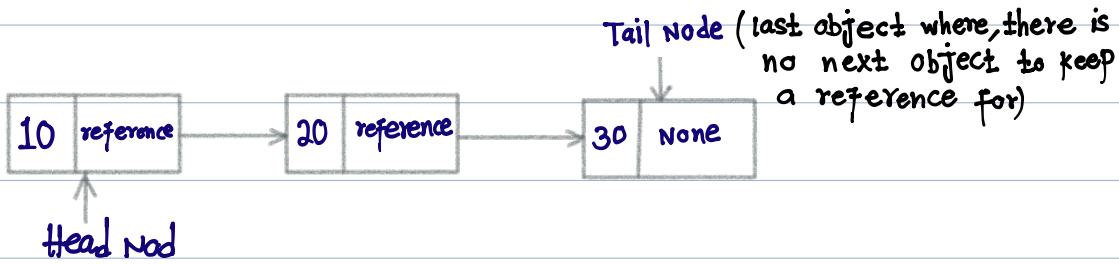
f has the address of Nothing

First Node (a) is known as head Node and last Node (f) is known as tail Node.



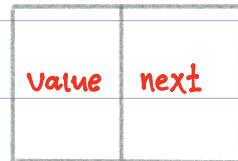
Python Implementation

Node : it can be represented using class in python, it has two fields, data field where the actual data is stored, and an address field where the reference to next node is stored.



class Node:

```
def __init__(self, value):  
    self.data = value  
    self.next = None
```



head = Node(10)



node1 = Node(20)

tail = Node(30)

head.next = node1



node1.next = tail



tail.next = None

