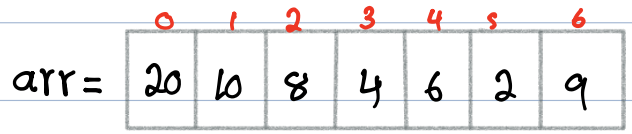
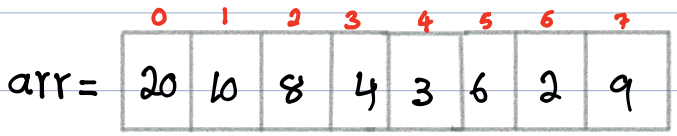
# Arrays Disadvantages

① size is fixed, if array is full and a new array to be created when inserting a new element, Time complexity for that operation becomes O(n).

② Insertion at the beginning and midle is costly




Insert (3) at 3rd index.



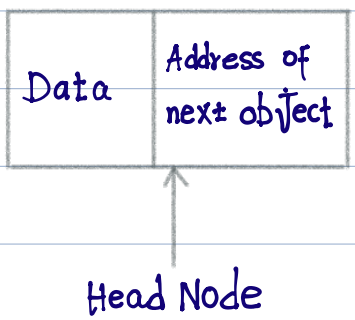

It can observed that insert one element in middle entire elements after mid have to be shifted by one index towards the right side

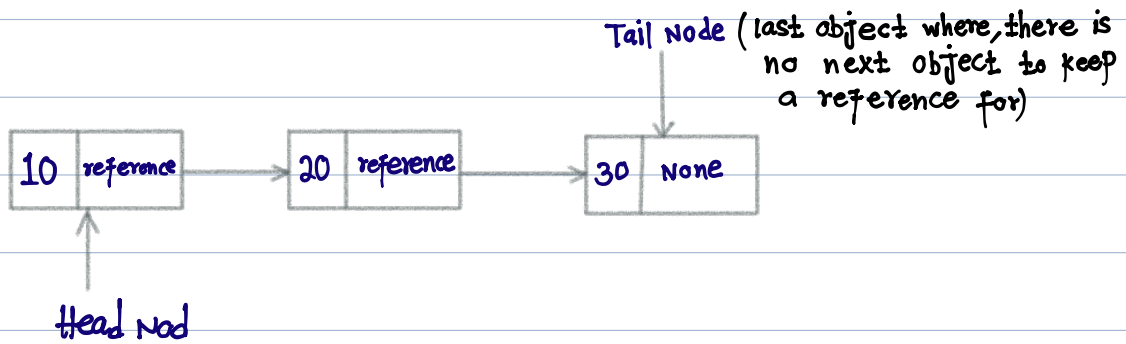
③ Implementation of queue & dequ data structures are defficult with arrays.

# Linked List

a linear data structure where, every object keep a reference the next object in the memory, those objects can be in completely different locations in memory.

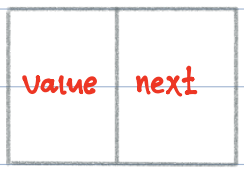
There is a special node called the head node, where the reference to 1st Node is kept, once head node is found, whole linked list can be traversed



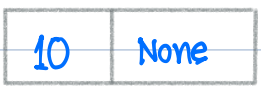



**Node :** it can be represented using class in Python, it has two fields, data field where the actual data is stored and an Address field where the reference to next node is stored

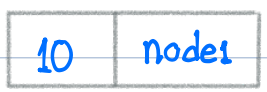
```
class Node:
    def __init__(self, value):
        self.data = value
        self.next = None
```



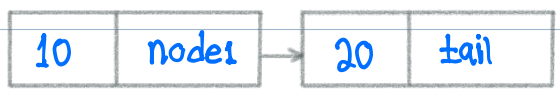

```
head = Node(10)
node1 = Node(20)
tail = Node(30)
```




```
head.next = node1
```




```
node.next = tail
```




```
tail.next = None
```