LAB MANAUAL 4B (ECE 3104/ECIE3101)

Linked List

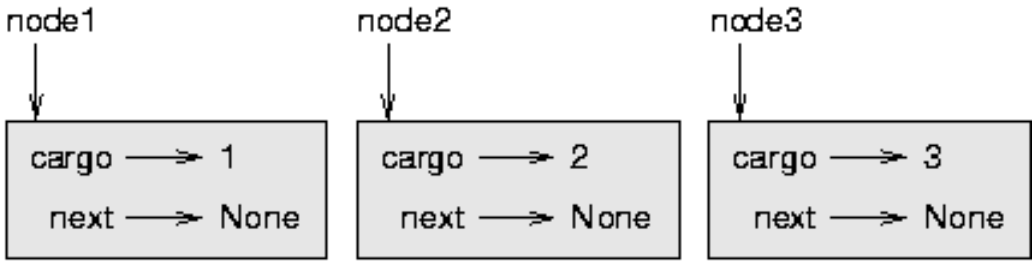
AIM: To write a program that implements the basic operations of the linked list in Python

Linked List Overview

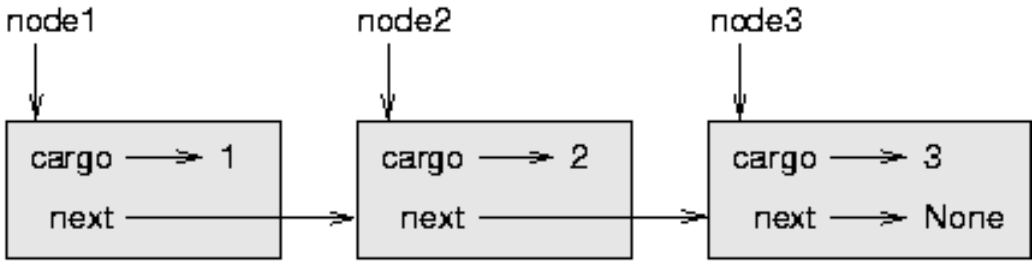
List is a collection of components, called nodes. Every node (except the last node) contains the address of the next node. Every node in a linked list has two components:

1. one to store the relevant information (that is, data)
2. one to store the address, called the link or next, of the next node in the list.

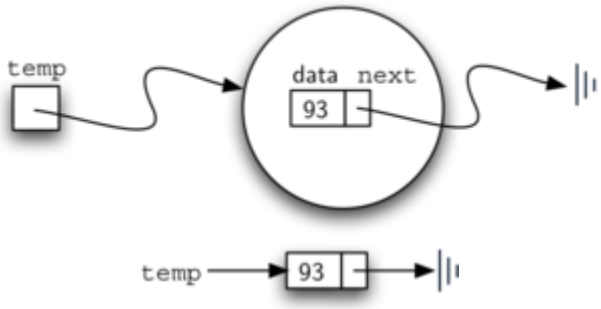
- ✓ The address of the first node in the list is stored in a separate location, called the head or first
- ✓ The address of the last node in the list is stored in a separate location, called the tail or last



Connecting nodes with each other.



To construct a node, you need to supply the initial data value for the node. Evaluating the assignment statement below will yield a node object containing the value 93



The Node class

The Node class also includes the usual methods to access and modify the data and the next reference.

SINGLY LINKED LIST

AIM:

To perform all the singly linkedlist operations.

Write a python program to perform the following operations:

a) Create a LList called myBucket
b) Delete an item from myBucket
c) Show the elements in the list
d) Check if list is empty

ALGORITHM STEPS

Step 1: Declare the functions to create, display, delete and check empty list. (use class to make it compact)
Step 2: Declare the variables in the main function.
Step 3: In a switch case get each functions number.
Step 4: To append the list created in the memory
Step 5: Assign a variable temp using pointers.

Step 6: To delete a node create a dummy variable.
Step 7: Check if the list is empty otherwise display the list using for statement.
Step 8: To insert a node in as first element

INPUT/OUTPUT

Singly Linked List Menu:
1. Create or Append List
2. Insert in Beginning
4. Remove from the List
5. Count element in the list
6. Display
7. Exit

Node Class definition

```
class Node(object):

       def __init__(self, d, n=None):
              self.data = d
              self.next_node = n
              # end of Class Note
       def get_next(self):
              return self.next_node
    def set_next(self, n):
       self.next_node = n
    def get_data(self):
       return self.data
    def set_data(self,d):
       self.data = d
```

LinkedList Class definition

```
class LinkedList (object):
        def __init__(self, r = None):
         self.root = r
         self.size  = 0
    def get_size (self):
        return self.size
```

Adding new node to the Llist

```
    def Add(self, d):
        new_node= Node(d, self.root)
        self.root=new_node
        self.size+=1
```

Search for node in the Llist

```
def Search(self,d):
        Curr_node = self.root
        while Curr_node:
                if Curr_node.get_data() == d:
                        return d
                else:
                        self.root= Curr_node.get_next()
                        return None
```

Lab Exercise 1(Individual)

Use the code given in the Lab section to create the following function:

1. Add a function that Insert in the Linkedlist
2. Modify Add a function that will enable you to Enter more items in insert function using (Y/N): E.g

   Enter your Choice: 1
   Enter number to add to list: 12
   Enter more(y/n): y
   Enter number to add to list: 13
   Enter more(y/n): n

3. Add a function (Count ()) that count total element added in Linkedlist
4. A function to Sort the nodes in the linked list
5. Delete from front/head of list
6. Delete from last/tail of list
7. check if the list is empty or not

The following in the menu

Singly Linked List Menu:
1. Create or Append List
2. Insert at Beginning
3. Sort the LList
4. Remove from the List (Head/Tail) H/T or you can write a separate function for it
5. Count Element in List
6. Show List of Elements
7. Search LList
8. Check if the list is empty or not
9. Exit

All the Singly Linked List operations are performed.

Submission on Sunday 8th October, 2017 before or by 11.59AM (mid night)