



Data Structure and Algorithm

Laboratory Activity No.7

Singly Linked List

Submitted by:

Manigbas, Jeus Miguel T. <Leader>

Barredo, Alwin P.

Bela, Lorenzo Miguel D.

Callorina, Robert Victor A.

Guzon, Kean Louiz I.

Instructor:

Engr. Maria Rizette H. Sayo

November, 22, 2023

I. Objectives

Introduction

A linked list is an organization of a list where each item in the list is in a separate node. Linked lists look like the links in a chain. Each link is attached to the next link by a reference that points to the next link in the chain. When working with a linked list, each link in the chain is called a Node. Each node consists of two pieces of information, an item, which is the data associated with the node, and a link to the next node in the linked list, often called next.

This laboratory activity aims to implement the principles and techniques in:

- Writing algorithms using Linked list
- Writing a python program that will perform the common operations in a singly linked list

II. Methods

- Write a Python program to create a singly linked list of prime numbers less than 20. By iterating through the list, display all the prime numbers, the head, and the tail of the list. (using Google Colab)
- Save your source codes to GitHub

III. Results

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

class LinkedList:
    def __init__(self):
        self.head = None
        self.tail = None

    def insert(self, new_data):
        new_node = Node(new_data)
        if self.head is None:
            self.head = new_node
            self.tail = new_node
        else:
            self.tail.next = new_node
            self.tail = new_node

primes = [2, 3, 5, 7, 11, 13, 17, 19]
list = LinkedList()
```

Figure 1.1: First Half of the Code

```

for prime in primes:
    list.insert(prime)

numb = list.head
prime_numbers = []
while numb:
    prime_numbers.append(str(numb.data))
    numb = numb.next

print("Prime numbers are:", ', '.join(prime_numbers))
print('Head:', llist.head.data)
print('Tail:', llist.tail.data)

Prime numbers are: 2, 3, 5, 7, 11, 13, 17, 19
Head: 2
Tail: 19

```

Figure 1.2: Second Half of the Code with the Output

This piece of code sets up a way to organize data using a linked list. Think of a linked list like a chain, where each link (node) holds some information and a connection to the next link. The code defines a node and a linked list, and the "insert" part is like adding a new link to the end of the chain. It's smart enough to know whether the chain is empty or not. Then, it uses this linked list idea to gather prime numbers, which are like special links, and showcases them. It's kind of like creating a collection of these special links and showing off the first and last ones for reference, just so you can see the whole picture of the chain.

The provided code sets up a linked list to organize data and then uses this concept to gather prime numbers and display them. The linked list is a data structure that consists of nodes linked together. Each node contains data and a reference to the next node. The code defines a Node class to represent a node and a LinkedList class to create and manipulate the linked list. The insert method adds a new node to the end of the list, and it handles both empty and non-empty lists. The code then uses the linked list to store prime numbers and showcases them. It creates a LinkedList object, inserts prime numbers into it, and then prints the prime numbers, as well as the head and tail of the linked list.

IV. Conclusion

To conclude, the linked list is a useful data structure when you want to easily insert items in between other items, and when you want to insert and remove elements without changing the index of all other items. It is particularly efficient for inserting and deleting elements, but accessing items in the list has a linear time complexity.

The provided code demonstrates the implementation of a singly linked list in Python, where each node has a link to the next node in the list. The Node class represents a node, and the

LinkedList class is used to create and manipulate the linked list. The insert method adds a new node at the head of the linked list, and the code showcases the usage of the linked list to store and display prime numbers.

References

[1] GeeksforGeeks. (2023a, September 26). *Introduction to Singly linked list*.

<https://www.geeksforgeeks.org/data-structures/linked-list/singly-linked-list/>

[2] *Singly Linked List Tutorials & Notes / Data Structures / HackerEarth*. (2016, April 24).

HackerEarth. <https://www.hackerearth.com/practice/data-structures/linked-list/singly-linked-list/tutorial/>

[3] Kumar, G. S. (2023, April 4). *Singly Linked List in Python with Examples*. Spark by

{Examples}. <https://sparkbyexamples.com/python/singly-linked-list-in-python-with-examples/>

[4] *Singly Linked List : Inserting, Appending and Freeing nodes*. :: *AlgoTree*. (n.d.).

https://www.algotree.org/algorithms/linkedlists/single_linked_list_insert_and_append/#:~:text=To%20insert%20into%20a%20singly,new_node%20is%20to%20be%20inserted.&text=Create%20a%20new_node%20that%20is%20to%20be%20inserted.