UNIVERSITY OF CALOOCAN CITY
COMPUTER ENGINEERING DEPARTMENT

Data Structure and Algorithm

Laboratory Activity No. 6

Singly Linked Lists

Submitted by:
Poliño, Justine

Instructor:
Engr. Maria Rizette H. Sayo

August 23, 2025

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.

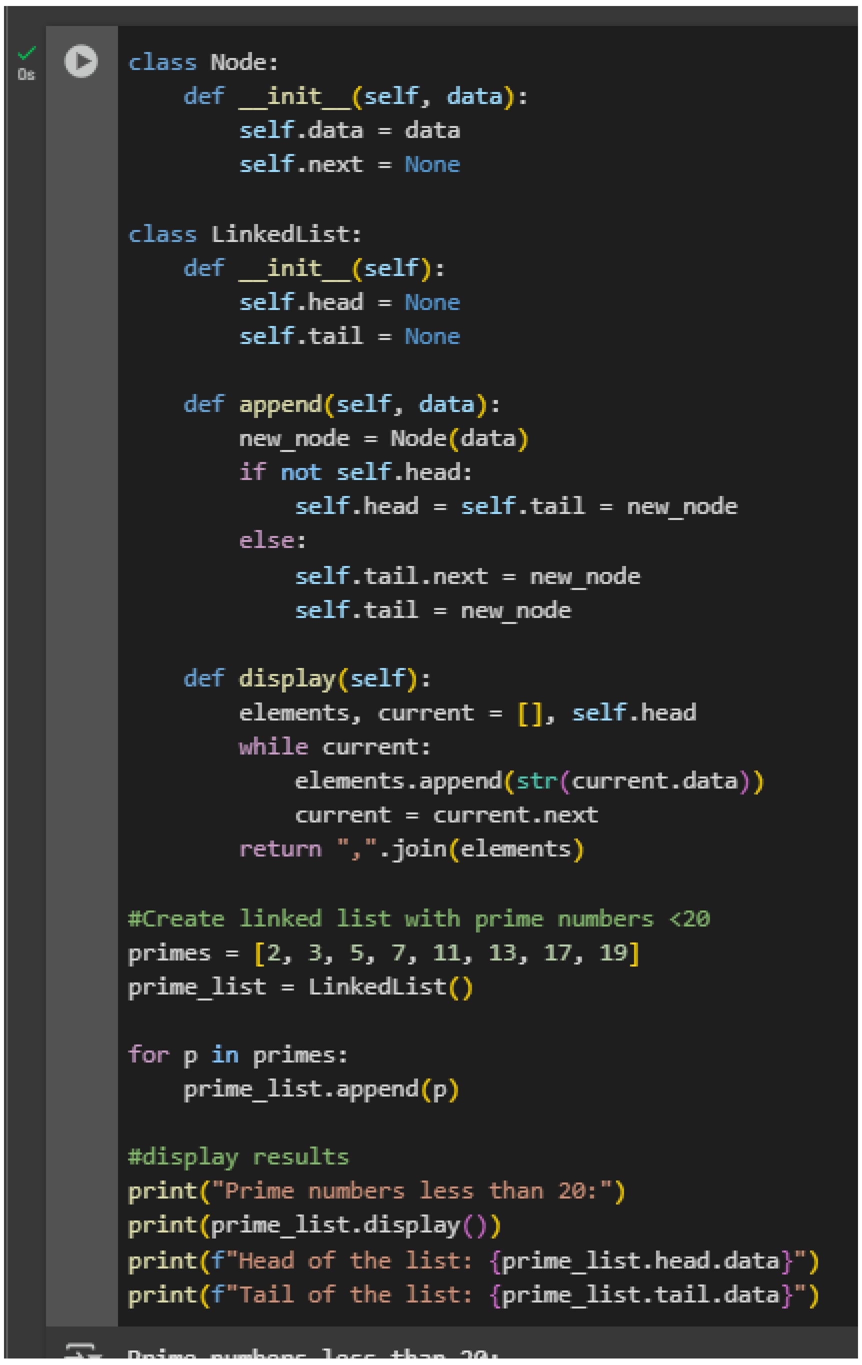
This laboratory activity aims to implement the principles and techniques in:

- 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

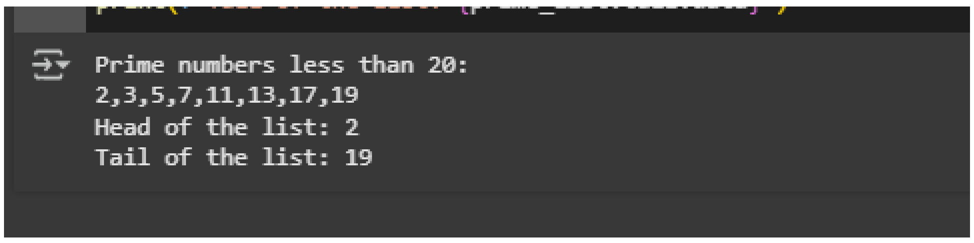
III. Results



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

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

    def append(self, data):
        new_node = Node(data)
        if not self.head:
            self.head = self.tail = new_node
        else:
            self.tail.next = new_node
            self.tail = new_node

    def display(self):
        elements, current = [], self.head
        while current:
            elements.append(str(current.data))
            current = current.next
        return ",".join(elements)

#Create linked list with prime numbers <20
primes = [2, 3, 5, 7, 11, 13, 17, 19]
prime_list = LinkedList()

for p in primes:
    prime_list.append(p)

#display results
print("Prime numbers less than 20:")
print(prime_list.display())
print(f"Head of the list: {prime_list.head.data}")
print(f"Tail of the list: {prime_list.tail.data}")
```

Figure 1 Screenshot of program from google colab

This Python program demonstrates you how to make and use a singly linked list. It uses a Node class, which contains both data and a pointer to the next node, to store prime numbers under 20. By adding each number consecutively and keeping track of the head and tail pointers, the implementation effectively constructs the list. A traversal function in the code illustrates the connections between nodes from the head (first element) to the tail (last element) by displaying each element in order.



```
Prime numbers less than 20:
2,3,5,7,11,13,17,19
Head of the list: 2
Tail of the list: 19
```

Figure 2 Screenshot of output from google colab

The result shown shows that a single linked list with prime integers under 20 can be implemented successfully. The entire traversal from beginning to end is displayed in the first line, which presents every element in the list as a connected sequence. The correct structure of the linked list, in which each node points to the next in the chain, is confirmed by the second and third lines, which explicitly identify the list's head (first element) & tail (last element).

IV. Conclusion

I learned how to create and manage a linked list using nodes with data and next pointers.I can now efficiently add elements and traverse through the entire list. This helps me understand how data connects sequentially in computer memory.

References

[1] Co Arthur O.. “University of Caloocan City Computer Engineering Department Honor Code,” UCC-CpE Departmental Policies, 2020.