UNIVERSITY OF CALOOCAN CITY
COMPUTER ENGINEERING DEPARTMENT

Data Structure and Algorithm

Laboratory Activity No. 6

Singly Linked Lists

Submitted by:
Mamanao, Kurt Marwin C

Instructor:
Engr. Maria Rizette H. Sayo

Aug, 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

Present the visualized procedures done. Also present the results with corresponding data visualizations such as graphs, charts, tables, or image . Please provide insights, commentaries, or explanations regarding the data. If an explanation requires the support of literature such as academic journals, books, magazines, reports, or web articles please cite and reference them using the IEEE format.

Please take note of the styles on the style ribbon as these would serve as the style format of this laboratory report. The body style is Times New Roman size 12, line spacing: 1.5. Body text should be in Justified alignment, while captions should be center-aligned. Images should be readable and include captions. Please refer to the sample below:

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

Figure 1: I define the basic building block of the linked list. Each node holds data and a reference to the next node.

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


    def append(self, data):
        new_node = Node(data)
        if not self.head:
            self.head = new_node
            return
        temp = self.head
        while temp.next:
            temp = temp.next
        temp.next = new_node


    def display(self):
        elements = []
        temp = self.head
        while temp:
            elements.append(temp.data)
            temp = temp.next
        return elements


    def get_tail(self):
        temp = self.head
        while temp and temp.next:
            temp = temp.next
        return temp.data if temp else None
```

figure 2: Here, I created the LinkedList class which manages the entire list. It includes methods to append new nodes, display the list, and get the tail node.

```python
def is_prime(n):
    if n < 2:
        return False
    for i in range(2, int(n**0.5) + 1):
        if n % i == 0:
            return False
    return True
```

Figure 3: This function checks if a number is prime. I used it to filter out prime numbers less than 20 for the linked list.

```python
prime_list = LinkedList()
for num in range(20):
    if is_prime(num):
        prime_list.append(num)
```

figure 4: In this part, I initialized the linked list and added all prime numbers below 20 using a loop and the “is_prime” function.

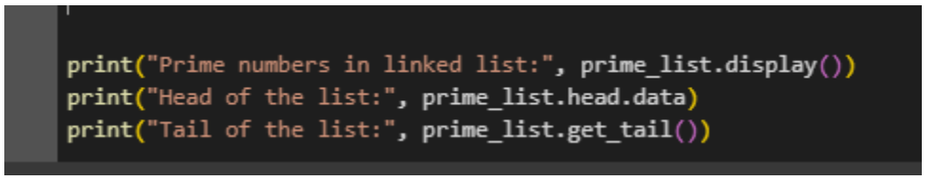

Figure 5: Finally, I displayed the contents of the linked list, including the head (first element) and tail (last element) to verify that everything was added correctly.

IV. Conclusion

In this activity, I successfully implemented a singly linked list in Python to store prime numbers less than 20. I demonstrated how to define a node structure, create and manage a linked list, and perform basic operations such as appending, displaying, and accessing the head and tail.

References

[1] M. Lutz, *Learning Python*, 5th ed. Sebastopol, CA: O'Reilly Media, 2013.