UNIVERSITY OF CALOOCAN CITY
COMPUTER ENGINEERING DEPARTMENT

Data Structure and Algorithm

Laboratory Activity No. 6

Singly Linked Lists

Submitted by:
Dispo, Lei Andrew T.

Instructor:
Engr. Maria Rizette H. Sayo

08, 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

- **ALGORITHM**

1. Start
2. Define a node structure
3. Define a function is_prime(n) to check if n is prime number
4. Initialize two pointers head= none, tails=none
5. Repeat for each num from 2 to 19
6. Print “Prime numbers:”
7. Traverse the linked list starting from head
8. Print value stored in head and tails
9. End

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

def is_prime(n):
    if n < 2:
        return False
    for i in range(2, int(n ** 0.5) + 1):  # corrected num -> n
        if n % i == 0:
            return False
    return True


head = None
tail = None

for num in range(2, 20):
    if is_prime(num):
        new_node = Node(num)
        if head is None:
            head = new_node
            tail = new_node
        else:
            tail.next = new_node
            tail = new_node


print("Prime numbers:")
current = head
while current:
    print(current.data, end=" ")
    current = current.next

print("\nHead:", head.data)
print("Tail:", tail.data)
```

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

IV. Conclusion

This report helps me reread linked list and learn how to use node and how to combine the both of them.

References

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