



UNIVERSITY OF CALOOCAN CITY
COMPUTER ENGINEERING DEPARTMENT



Data Structure and Algorithm

Laboratory Activity No. 6

Singly Linked Lists

Submitted by:
Bron, Jhustine A.

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

```
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):
        nnode = Node(data)
        if self.head is None:
            self.head = nnode
            self.tail = nnode
        else:
            self.tail.next = nnode
            self.tail = nnode

    def display(self):
        current = self.head
        while current:
            print(current.data, end=" ")
            current = current.next
        print()
```

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

Lili = LinkedList()

for i in range(20):
    if is_prime(i):
        Lili.append(i)

print("Prime numbers from 1-20:")
Lili.display()

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

Prime numbers from 1-20:
2 3 5 7 11 13 17 19

Head: 2
Tail: 19

Figure 1. Screenshot of the Program

Insights:

This program creates a singly linked list that stores all prime numbers less than 20. The Node class represents each element, while the LinkedList class manages the head, tail, and appending of new nodes. After inserting primes, the list displays 2 3 5 7 11 13 17 19, with 2 as the head and 19 as the tail. This shows how linked lists can organize data in sequence while using pointers to connect nodes.

IV. Conclusion

In this laboratory, I was able to create a singly linked list that stores all prime numbers less than 20. Through this task, I learned how to build nodes, connect them using pointers, and keep track of both the head and tail of the list. Seeing the output 2 3 5 7 11 13 17 19 confirmed that my code worked correctly. Overall, this activity helped me better understand how linked lists function and why they are useful in organizing data.