



UNIVERSITY OF CALOOCAN CITY
COMPUTER ENGINEERING DEPARTMENT



Data Structure and Algorithm

Laboratory Activity No. 6

Singly Linked Lists

Submitted by:
Regondola, Jezreel P.

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 = None):
        self.data = data
        self.next = None

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

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

    def display(self):
        current = self.head
        while current:
            print(current.data, end = ", ")
            current = current.next
        print("None")

    def get_head(self):
        if self.head:
            return self.head.data
        else:
            return None

    def get_tail(self):
        if self.tail:
            return self.tail.data
        else:
            return None

def get_primes(limit):
    primes = []
    for num in range(2, limit):
        is_prime = True
        for i in range(2, int(num ** 0.5) + 1):
            if num % i == 0:
                is_prime = False
                break
        if is_prime:
            primes.append(num)
    return primes

primes = get_primes(20)
linked_list = SinglyLinkedList()

for prime in primes:
    linked_list.append(prime)

print("Linked list prime numbers:")
linked_list.display()

print(f"Head of the list: {linked_list.get_head()}")
print(f"Tail of the list: {linked_list.get_tail()}")
```

→ Linked list prime numbers:
2, 3, 5, 7, 11, 13, 17, 19, None
Head of the list: 2
Tail of the list: 19

Figure 1 Screenshot of program

This program generates prime numbers less than 20, stores them in a singly linked list, and then displays the list along with the head (first prime) and tail (last prime).

IV. Conclusion

This activity helped us implement and understand the core principles of linked lists. By writing a Python program to create a singly linked list of prime numbers, we learned how to perform a basic linked list operations such as insertion, traversal, and accessing the head and tail.

References

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