Data Structure and Algorithm

Laboratory Activity No. 6

Singly Linked Lists

Submitted by:
Palmes, Lewis Clark L.

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.
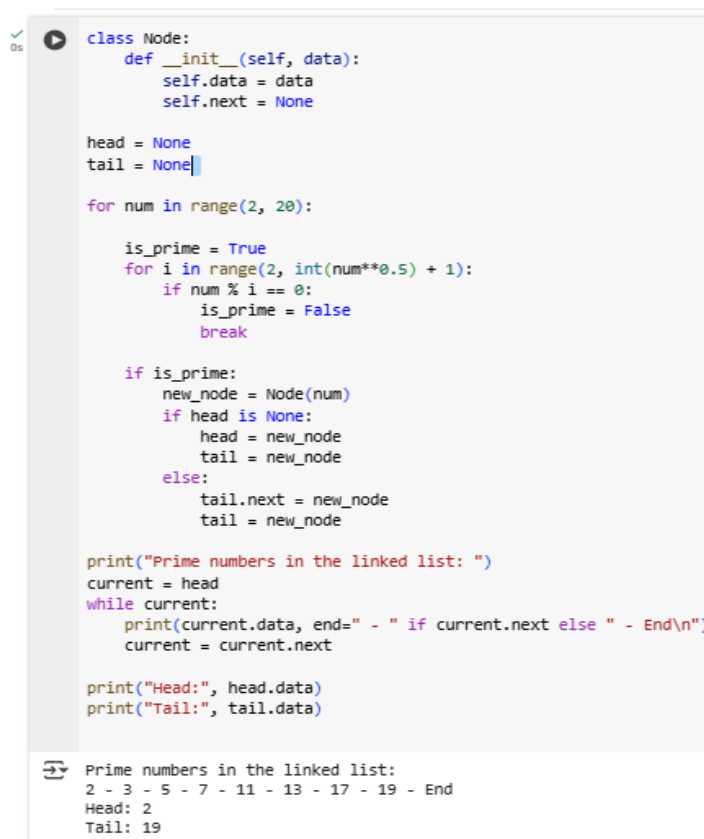
This laboratory activity aims to implement the principles and techniques in:
- Writing algorithms using Linked list
- Writing a python program that will perform the common operations in a singly linked list

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

head = None
tail = None

for num in range(2, 20):

    is_prime = True
    for i in range(2, int(num**0.5) + 1):
        if num % i == 0:
            is_prime = False
            break

    if is_prime:
        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 in the linked list: ")
current = head
while current:
    print(current.data, end=" - " if current.next else " - End\n")
    current = current.next

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

```
Prime numbers in the linked list:
2 - 3 - 5 - 7 - 11 - 13 - 17 - 19 - End
Head: 2
Tail: 19
```

Figure 1 Screenshot of program

1

This Python program creates a singly linked list containing all prime numbers less than 20. It first defines a Node class to represent each element in the list, with data and a pointer to the next node. The program checks each number from 2 to 19 to determine if it is prime, and if so, creates a new node for that number. The first prime found becomes the head and tail of the list, while subsequent primes are appended to the end by updating the next pointer of the tail. Finally, the program prints all prime numbers in the linked list, along with the values of the head and tail nodes.

IV. Conclusion

This laboratory activity showed how a singly linked list can be created and traversed in Python using prime numbers less than 20. By using nodes with data and pointers, it became clearer how linked lists store and connect elements step by step. The activity helped deepen the understanding of basic linked list operations like creating, inserting, and traversing nodes, highlighting their importance as a foundation in learning data structures.

References

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

[2] "Google Colab." https://colab.research.google.com/drive/1Cxe7CgpyPag6uZNEjA9psxMV25VfNLIE#scrollTo=bfJEBslNMK V9

[3] Lewis-Clark-Palmes, "CPE-201L-DSA-2-A/LAB_6.ipynb at main · Lewis-Clark-Palmes/CPE-201L-DSA-2-A," *GitHub*. https://github.com/Lewis-Clark-Palmes/CPE-201L-DSA-2-A/blob/main/LAB_6.ipynb

[4] GeeksforGeeks, "Singly linked list tutorial," *GeeksforGeeks*, Jul. 23, 2025. https://www.geeksforgeeks.org/dsa/singly-linked-list-tutorial/