UNIVERSITY OF CALOOCAN CITY
COMPUTER ENGINEERING DEPARTMENT

Data Structure and Algorithm

Laboratory Activity No. 6

Singly Linked Lists

Submitted by:
Uy, Junichiro H.

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.

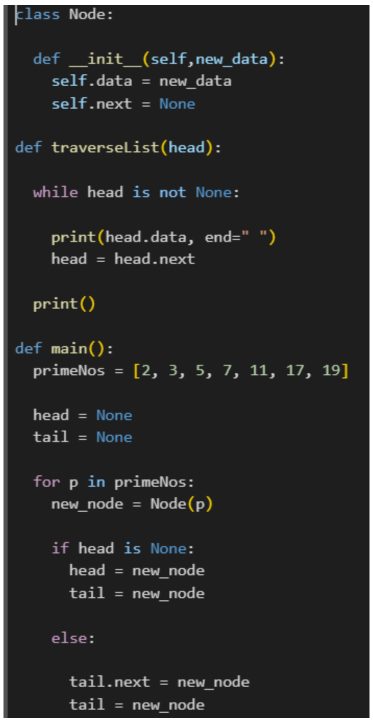
This laboratory activity aims to implement the principles and techniques in:

- 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

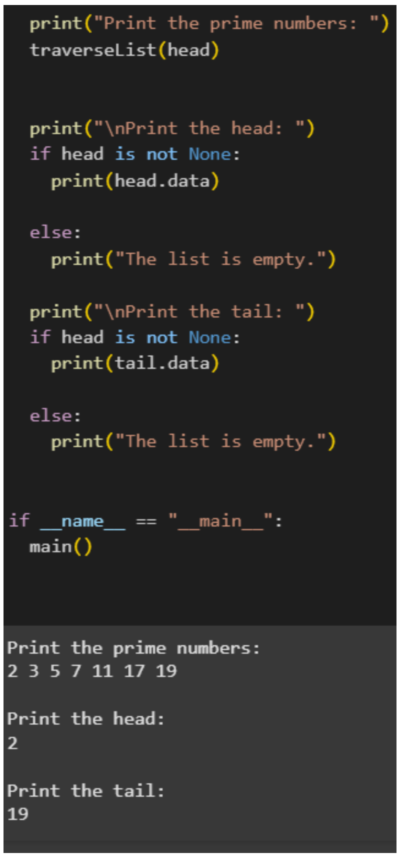
III. Results



```
class Node:

  def __init__(self,new_data):
    self.data = new_data
    self.next = None

def traverseList(head):

  while head is not None:

    print(head.data, end=" ")
    head = head.next

  print()

def main():
  primeNos = [2, 3, 5, 7, 11, 17, 19]

  head = None
  tail = None

  for p in primeNos:
    new_node = Node(p)

    if head is None:
      head = new_node
      tail = new_node

    else:

      tail.next = new_node
      tail = new_node
```

Figure 1 Screenshot of program from colab

This program shows how to make a singly linked list using the class node. With the instructors being data and next. I also added a function traversal, so it prints each heads progressing. Then I initialized the prime numbers list, the head, and the tail. Then I made a for loop that takes each element inside the list and puts them in a singly linked list format, making the code much shorter. I also put the current element inside the head value and tail value so we can print it later.



```
  print("Print the prime numbers: ")
  traverseList(head)


  print("\nPrint the head: ")
  if head is not None:
    print(head.data)

  else:
    print("The list is empty.")

  print("\nPrint the tail: ")
  if head is not None:
    print(tail.data)

  else:
    print("The list is empty.")


if __name__ == "__main__":
  main()
```

```
Print the prime numbers:
2 3 5 7 11 17 19

Print the head:
2

Print the tail:
19
```

Figure 2 Screenshot of program and output from colab

This image shows how I used the traverse function then how I printed the head and tail variable that we initialized earlier. I also used an if and else to output when the list is empty.

IV. Conclusion

I learned how to make singly linked list using a class node. My code uses a simple structure of many singly linked lists. I also used a for loop to do it quicker.

References

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