

Homework 1

Typeset solutions only (Latex) in pdf.

Problem 1: Determine analytically a closed form expression for the total number of times an asterisk is printed in the following Python code:

```
n = int(input('Enter a number n: '))

for i in range(n):
    for j in range(i+1):
        print('*', end = '')
    print()
```

Problem 2: Assume a single-linked list that has an ordered insert operation where we insert before a node that has larger content. We also have an insert_tail and insert_head operation.

```
class Node:
    def __init__(self, content):
        self.content = content
    def __str__(self):
        return f'N[{self.content}]'

class Single_Linked_List:

    def __init__(self):
        self.head = None

    def __str__(self):
        result = []
        current = self.head
        while current:
            result.append(str(current))
            current = current.next
        return ' '.join(result)

    def insert_tail(self, content):
        if not self.head:
            self.head = Node(content)
            self.head.next = None
            return
        current = self.head
        while current.next:
            current = current.next
        current.next = Node(content)
        current.next.next = None

    def insert_ordered(self, content):
        if not self.head:
```

```

        self.head = Node(content)
        self.head.next = None
        return
    if content < self.head.content:
        newNode = Node(content)
        newNode.next = self.head
        self.head = newNode
        return
    current = self.head
    while current.next and current.next.content < content:
        current = current.next
    if not current.next:
        current.next = Node(content)
        current.next.next = None
        return
    after_node = current.next
    current.next = Node(content)
    current.next.next = after_node
    return

def insert_head(self, content):
    new_head = Node(content)
    new_head.next = self.head
    self.head = new_head.next

```

- (A) How many (existing) nodes are accessed if we use `insert_tail` and the list has already n elements?
- (B) What is the minimum, the maximum and the average number of existing nodes accessed if we use `insert_ordered` with a list which has already n elements.
- (C) The current implementation of `insert_head` **does not work**. Find the error in the code and correct it.