HW 0

Get your Python skills up to date with the following exercises.

Q1

Find the indices of the two numbers such that they add up to target given an array of integers nums and a target integer. Assume that each input has exactly one solution, and do not use the same element twice. Answers can be returned in any order.

Example:
```
Input: nums = [2,8,11,15], target = 10
Output: [0,1]
```

Signature of the function:
```
twoSum(nums: list[int], target: int) -> List[int]
```

Can you come up with an algorithm that is less than O(n^2) time complexity?

Q2

In the array "prices", prices[i] represents the price of a stock on the ith day. To maximize your profit, choose a day for buying one stock, and another day for selling it. Calculate the maximum profit you can make from this transaction. Return 0 if you are unable to achieve any profit.

Example:
```
Input: prices = [7,1,5,3,6,4]
Output: 5
(Buy on day 2 (price = 1) and sell on day 5 (price = 6), profit = 6-1 = 5)
```

Q3

A linked list is a basic data structure for expressing a sequence of nodes. In the Linked List, the first element is the head and the last element is the tail. Each node in the sequence has a pointer to the next node. The following is an implementation of a node in a linked list.

```
class Node:
    def __init__(self, value, next_node=None):
        self.value = value
        self.next = next_node
```

```
    def __str__(self):
        return str(self.value)
```

Q3.1

Write a Python module that reads a file containing a list of integers separated by a semicolon and returns the head node of a linked list.

Signature of the function:
`read_file(file_path: str) -> Node`

Q3.2

Write a module that gives the length of a given linked list.

Signature of the function:
`get_length(head: Node) -> int`

Q3.2

Write a module that sorts a given linked list in place, i.e., without copying its contents.

Signature of the function:
`sort_in_place(head: Node) -> Node`

What is the time complexity of your method?
What is the memory complexity of your method?