FIT1008 Introduction to Computer Science (FIT2085 for Engineers)

Tutorial 9

Semester 1, 2019

Objectives of this tutorial

- To understand recursion.
- To understand quicksort and merge sort.

Exercise 1 *

Consider a Node class which defines a node for a linked data structure, and which is defined as follows:

```
class Node:
        def __init__(self, item = None, link = None):
                self.item = item
                self.next = link
```

Suppose you have a List class that implements a Linked List using the Node class above, and has the following method.

```
def mystery(self):
        return mystery_aux(self.head)

def mystery_aux(self, current):
        if current == None:
                return 0
        else:
                current.item +=  mystery_aux(current.next)
                return current.item
```

(a) What does the mystery method do? Explain in terms of its effect on the value of a_list, that consists of the following items in order 1,2,3,4,5.

(b) What is the best and worst complexity in Big O notation of our mystery() method in terms of the length of the list (N)?

(c) How would you define the method iteratively?

Exercise 2 *

(a) Write a *recursive* method for computing the sum of the digits of a number. For example, for number 979853562951413, the sum of its digits is $9+7+9+8+5+3+5+6+2+9+5+1+4+1+3 = 77$. To do this you can use integer division by 10 (//10) which returns an integer with the same digits except the last one, and reminder by 10 (% 10), which returns the last digit. For example, if you have X = 3456, then X//10 gives you 345, while X%10 gives you 6.

(b) Determine its complexity, in Big-O notation.

Exercise 3 *

In Quicksort, the choice of pivot is crucial. Discuss the reasons for this and give some examples of good/bad choices

Exercise 4 *

Are Mergesort, or Quicksort stable? Discuss and provide examples.

Exercise 5

Definition: The *digital root* of a decimal integer is obtained by adding up its digits, and then doing the same to *that* number, and so on, until you get a single digit, which is the digital root of the number you started with.

For example, to find the digital root of 979853562951413, we calculate: sum of digits $= 9+7+9+8+5+3+5+6+2+9+5+1+4+1+3 = 77$, then sum of digits $= 7+7 = 14$, then sum of digits $= 1+4 = 5$. Now we have just one digit, 5, so that's the digital root of the number we started with.

(c) Write a *recursive* method to compute the digital root of a positive integer.

(d) Determine its complexity, in Big-O notation.