National University of Singapore
School of Computing
IT5003: Data Structure and Algorithm
Semester I, 2019/2020

Special Practice Questions
Recursion

Although you can give pseudo code for the following questions, it is easier (and better practice) if you express it in Python.

1. [Recursion on Number] Give a recursive function *logR*(N, B) that takes in a number N and a base B, both positive (i.e. > 0). The function returns $\lfloor log_B N \rfloor$. e.g. *logR*(7, 2) = 2, *logR*(1234, 10) = 4.

2. [Recursion on List] Give a recursive function *filter*(L, target) that takes in a Python List L (containing numbers) and a target number. This function removes all copies of target number from the list L and return L. e.g. *filter*([3, 1, 3, 2, 6, 3], 3) ➔ [1, 2, 6] ; *filter*([3, 1, 3, 2, 6, 3], 5) ➔ [3, 1, 3, 2, 6, 3]

3. [Recursion on Linked List] Give a recursive function *countIf*(R, target) that takes in a Singly Linked List R and a target number. This function returns the number of occurrences of target in the list R. You can assume we used the SinglyNode class to build the linked list.

   e.g. Suppose R is a linked list with 5 → 2 → 1 → 13 → 2 → 9 → 2, then *countIf*(R, 2) ➔ 3; *countIf*(R, 7) ➔ 0

As mentioned in lecture, you need to "rewire your brain" to use recursive effectively. ☺ Do not discuss / search for answer, as understanding from solution is actually counter-productive. You'll lose the chance to understand and apply the thinking process behind a recursive solution. So, take some time and figure out the answers by yourself.

We will discuss 1-2 of the questions above as "pre-lecture" question.