FIT2004 S1/2017: Lab questions for week 2

Arun Konagurthu

Objectives: This prac is primarily designed to brush up your (i) programming basics and (ii) revise the basic concepts of iteration, recursion etc. you have learnt in FIT1029/FIT1045 (or equivalent) and FIT1008.

NOTE: This prac is **NOT** assessed. Write your programs in Python programming language.

1. Write a program to compute the factorial of a number recursively.
2. Write a iterative version of the same.
3. Write a program to print all prime numbers between 1 and N. Explore more efficient ways in doing the same.
4. Write a program to compute the *greatest common divisor* (GCD) of two positive numbers.
5. Write a program to multiply two matrices.
6. The Fibonacci series are numbers that appear in the following sequence:

   $$1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, \ldots$$

   Given some N as a command line parameter, write a program to compute any Nth number in the sequence. Can you reason the time-complexity of your implementation?
7. There is an $O(\log(N))$ time algorithm to solve this problem! (See non-examinable hand-written notes under week 2 section)

If you have managed to cruise through the above preparatory programming questions and still have time left in your lab to attempt more, consider the following task. (If you are unable to finish this in the lab, attempt this as an homework exercise.)

8. Your task is to implement the list abstract data type (ADT) in Python. (Refer to Week 1 slides). Your program should also implement its basic list operations: `null(...)`, `head(...)`, `tail(...)`, `length(...)`, `append(...)`, `merge(...)` discussed in the lecture.[1]

9. Time for a puzzle: In many traditions of metered poetry, a verse (or a metrical line) is written in some rhythmic pattern that is composed of a combination of `light` and `heavy` syllables. It is common to associate a weight of 1 unit with each `light` syllable, and a weight of two units with each `heavy` syllable. Define the `verse-weight` as the sum total weight of individual syllables in that verse. Write a program that can compute the total *number* of possible patterns involving light and/or heavy syllables given any `verse-weight` of W. What is the time complexity of your algorithm?

   As an example, a verse with weight of $W = 5$ can be composed of patterns such as `light-light-heavy-light` (1+1+2+1=5), or `heavy-light-heavy` (2+1+2=5), or any one of the six other possible patterns. So, if the input to your program were $W = 5$, the output would be $nPatterns = 8$.

-=oOo=-
END
-=oOo=-

[1]Note: when dealing with `merge(...)` operation, the two lists (passed as arguments) are assumed to be sorted. So consider adding a simple sorting functionality to your ADT – choose any of the sorting algorithms you have encountered in FIT1008/FIT1029/FIT1045.