Data Structures
Homework #2

Due: Oct 14, 2025

1. Describe a recursive algorithm to compute the integer part of the base-two logarithm of n using only addition and integer division where $n \geq 1$. Please give your idea first and then use pesudo-code to present the algorithm.

2. Give a recursive algorithm to compute the product of two positive integers, n and m, using only addition and subtraction.

3. Conider the Chebyshev polynomials of the first kind $\{T_n\}$ defined as following recursively.

$$
\begin{aligned}
&T_0(x) = 1 \\
&T_1(x) = x \\
&T_n(x) = 2 * x * T_{n-1}(x) - T_{n-2}(x), \text{for } n \geq 2
\end{aligned}
$$

   We now would like to list all the polynomials sequentially from degree 0 to some given degree n. In order to simplify the presention for a polynomial, one may use the coefficients of a polymial as a list (or an array). For example, we use list[4, 2, 3, 1] to represent the associated polynomial $p(x) = 4x^3 + 2x^2 + 3x + 1$. Please have the pseudo-code for the following finctions.

   (1) Please have the recursive function, `chebyshev_recursive()` to list all the polynomials with the recursive definition.

   (2) Like the inefficiency of the binary recursion method for Fibonacci numbers, we may use memoization approach we mentioned in class to avoid the redundant computation. Please provide the memoized recursive function, `chebyshev_memoreccu()` to list all the polynomials.

   (3) To avoid the redundant computation, one can also use the way we introduced as LinearFibonacci(). By referring to that, please have a linear recursive function `chebyshev_addoutrec()`.

   (4) Last, please provide an iterative (no recursive call) function, `chebyshev_iterative()`, which lists all the polynomials iteratively.

4. Suppose you would like to have a parking system for managing parking lots (or garages). Please design and provide an *abstract data type* (ADT) for a parking system. You need to give the definitions to the related objects and what operations are supported. Note that an ADT specifies what operations are supported, not how they are implemented. The goal is to capture the essential behavior of a parking system, independent of the underlying data structures.

5. (**Programming problem 1**)
   Consider the recursive approach in above problem 2.

   (1) Implement the approach as a function named as `product_rec` using Python.

(2) Please have an iterative version for the approach and write a function for this version with function name `product_ite`.

(3) Compare these two functions with the same input in terms of running time and write what you have observed.

6. (**Programming problem 2**)
Consider problem 3, the problem of Chebyshev polynomials of the first kind. Please use Python to implement the pseudo-code you provide for solving the problem. You should provide all the functions with the mentioned names, `chebyshev_recursive()`, `chebyshev_memoreccu()`, `chebyshev_addoutrec()`, and `chebyshev_iterative()` respectively. Then, perform a comparison on these functions by measuring the execution time.

   (1) Please have the recursive function, `chebyshev_recursive()` to list all the polynomials with the recursive definition. .

   (2) Please provide the memoized recursive function, `chebyshev_memoreccu()` to list all the polynomials.

   (3) Please use the way LinearFibonacci() we introduced in class to have a linear recursive function `chebyshev_addoutrec()`.

   (4) Please provide an iterative (no recursive call) function, `chebyshev_iterative()`, which lists all the polynomials iteratively.

   (5) In order to have the experiments for comparisons, please execute all the functions with the same input many times and then measure the practical running timme in average. Please write down your observations on these four versions.

Note: We will use an in-built python library `timeit` and the module function `timeit.timeit()` for measuring the running time respective. This will be provided in the template. For the programming problems, please do some experiments by yourself to observe the running time and the number of recursive calls. You need to report what you have observed from the experiments when you submit your homework.

About submitting this homework

1. For problem 1, 2, 3, and 4, Please

   (1) **write by hand** all of your solutions on the papers of size A4,

   (2) leave you name and student ID on the first page, and

   (3) hand in your solutions on papers to me in class

2. For problem 5 and 6, please upload the completed `.ipynb` file with the filename as `HW2_studentID.ipynb` to **i-school(Plus)** (`https://istudy.ntut.edu.tw/learn/index.php`).

3. The **deadline** is **midnight of Oct 14**, 2025 and **Late work** is not acceptable.