DISCRETE STRUCTURES
Lab 5
Sequences and Recursion

1. Introduction

In this tutorial, we'll practice sequences and recursion.

2. Tutorial

Read following lecture notes of Discrete Structures on site elit.tdtu.edu.vn

Week6_7_Sequences_and_Recursion.pdf

3. Exercise

1. We can define the sum from 1 to x recursively as follows for integer x ≥ 1:

$$\begin{cases} 1 & if\ x = 1 \\ x + sum\ from\ 1\ to\ x - 1 & if\ x > 1 \end{cases}$$

Write a Python program to compute the sum 1 + 2 + 3 + 4 + 5 + 6 + 7 recursively.

2. Write a recursive Python program so that it asks the user for which Fibonacci number he or she wants. Use your program to compute the 10th, 20th, 30th and 40th Fibonacci numbers. Why does it take so much longer to computer the higher Fibonacci numbers?

3. We can determine how many digits a positive integer has by repeatedly dividing by 10 (without keeping the remainder) until the number is less than 10, consisting of only 1 digit. We add 1 to this value for each time we divided by 10. Here is the recursive algorithm:

$$\begin{cases} If\ n\ <\ 10\ return\ 1. \\ Otherwise, return\ 1\ +\ the\ number\ of\ digits\ in\ n//10 \end{cases}$$

Implement this recursive algorithm in Python and test it using a main function that calls this with the values 15, 105, and 15105.

4. Write a recursive Python function that has a parameter representing a list of maximums is either the first value in the list or the maximum of the rest of the list, whichever is larger. If the list only has 1 integer, then its maximum is this single value, naturally.

5. Write a recursive Python function that finds and returns the minimum element in an array, where the array and its size are given as parameters.

6. Define a recursive Python function $P(n, r)$ such that

$$P(n, r) = \begin{cases} n(n-1)(n-2) \ldots (n-r+1) & if\ n \geq r > 0 \\ 1 & otherwise \end{cases}$$

7. Implement a Python solution to the Tower of Hanoi with n disks.

8. Reformulate Euclid' *gcd* algorithm as a recursive Python function

4. References:

http://www.cs.cmu.edu

http://www.bowdoin.edu/

http://interactivepython.org/