CPSC-354 Report

Natalie Huante
Chapman University

September 17, 2023

Abstract

Short summary of purpose and content.

Contents

1 Introduction

2 Homework

This section contains solutions to homework.

2.1 Week 1

The homework for Week 1 is dedicated to allow myself the opportunity to get familiar with LaTeX as well as review the model of equational reasoning. In this lesson, we used the Fibonacci Sequence as an example of this but we will use the function of Greatest Common Divisor for the assignment. In terms of familiarity with LaTeX, I do have experience through the Algorithm Analysis report from the previous semester, however, this serves as a way to remind myself of the language.

For context, the definition the GCD function is as follows:

```
gcd(a,b):
Input: Two whole numbers (integers) called a and b, both greater than 0.
(1) if a>b then replace a by a-b and go to (1).
(2) if b>a then replace b by b-a and go to (1).
Output: a
```

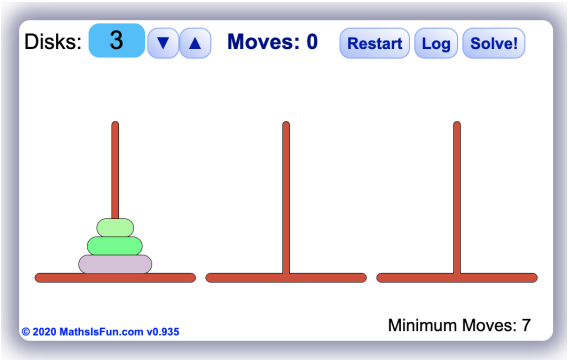
Now, I will write out the full computation of gcd(9,33) below:

$$
\begin{aligned}
gcd(9, 33) &= gcd(9, 24) \\
&= gcd(9, 15) \\
&= gcd(9, 6) \\
&= gcd(3, 6) \\
&= gcd(3, 3) \\
&= 3
\end{aligned}
$$

As you can see above, the basis of the function is to iteratively decrement a and b until you reach a point at which you can no longer decrement them by the definition given.

2.2 Week 2

For this week's homework we wil take a look and practice based on our discussion of the Towers of Hanoi problem. To solve the Towers of Hanoi one must move all the rings from the left-most tower to the right-most tower. However, discs can only be placed on other discs larger than themselves. Therefore, we have to logically think about the order in which we move them as well as how we can most efficiently do so (the least amount of moves). I have included a visual representation of what this problem looks like:



In our class discussion, we established a set of rules in order to then discuss how to represent our functions in code. Below I include both rules as well as repeat some notes on notation in order to allow for some context on the second half of this assignment.

```
hanoi 1 x y = move x y

hanoi (n+1) x y =
  hanoi n x (other x y)
  move x y
  hanoi n (other x y) y
```

- read `hanoi n x y` as "move tower of n disks from x to y"
- think about how to move a tower of n+1 disks assuming we already know how to move a tower of n disks
- `hanoi n x y` is a function that takes three arguments: a number n (the number of disks), a number x (encoding the place where the tower is, a number y (encoding the place where the tower should go))
- `move x y` is afunction that moves one disk (the topmost disk) from x to y

- `other x y` denotes the third place which is neither x nor y

Now having laid down some rules for our discussion, we observed that the nature of the Towers of Hanoi problem is recursive given that in order to solve a tower of 4 disks we must first solve it with 3 disks and then that requires we solve it for 2 disks and so on. In class we observed the 5 disk tower solution of `hanoi 5 0 2`. For this assignment, I will write out the implementation of this function to cement what we learned about its recursion and logic.

```
hanoi 5 0 2
  hanoi 4 0 1
    hanoi 3 0 2
      hanoi 2 0 1
        hanoi 1 0 2 = move 0 2
        move 0 1
        hanoi 1 2 1 = move 2 1
      move 0 2
      hanoi 2 1 2
        hanoi 1 1 0 = move 1 0
        move 1 2
        hanoi 1 0 2 = move 0 2
    move 0 1
    hanoi 3 2 1
      hanoi 2 2 0
        hanoi 1 2 1 = move 2 1
        move 2 0
        hanoi 1 1 0 = move 1 0
      move 2 1
      hanoi 2 0 1
        hanoi 1 0 2 = move 0 2
        move 0 1
        hanoi 1 2 1 = move 2 1
  move 0 2
  hanoi 4 1 2
    hanoi 3 1 0
      hanoi 2 1 2
        hanoi 1 1 0 = move 1 0
        move 1 2
        hanoi 1 0 2 = move 0 2
      move 1 0
      hanoi 2 2 0
        hanoi 1 2 1 = move 2 1
        move 2 0
        hanoi 1 1 0 = move 1 0
    move 1 2
    hanoi 3 0 2
      hanoi 2 0 1
        hanoi 1 0 2 = move 0 2
        move 0 1
        hanoi 1 2 1 = move 2 1
      move 0 2
      hanoi 2 1 2
        hanoi 1 1 0 = move 1 0
```

```
move 1 2
hanoi 1 0 2 = move 0 2
```

As we can see this is a very lengthy program that relies on its recursive nature to solve the hanoi problem. Now, if we extract from this execution the moves that solve the puzzle (in their right order), we will see that there are 31 moves for the 5-disk tower that will solve the problem most efficiently. I say that these are the steps that actually solve the problem since they are the ones that will prompt moving a disk from one tower to another. I will rewrite the steps again below:

```
move 0 2
move 0 1
move 2 1
move 0 2
move 1 0
move 1 2
move 0 2
move 0 1
move 2 1
move 2 0
move 1 0
move 2 1
move 0 2
move 0 1
move 2 1
move 0 2
move 1 0
move 1 2
move 0 2
move 1 0
move 2 1
move 2 0
move 1 0
move 1 2
move 0 2
move 0 1
move 2 1
move 0 2
move 1 0
move 1 2
move 0 2
```

Now that we have written out the exection, we can also see that the word `hanoi` appear many times in the computation. I observed the number of occurrences and recorded them in the below table. By looking at the table, we can notice that the number of occurrences can actually be represented as a formula, for they increase in the same intervals exponentially. Therefore, we can say that with n disks, the number of times the word `hanoi` appears in the computation is $2^n - 1$.

n disks	num of hanoi
1	1
2	3
3	7
4	15
5	31

2.3 Week 3

...

3 Conclusions

(approx 400 words) A critical reflection on the content of the course. Step back from the technical details. How does the course fit into the wider world of software engineering? What did you find most interesting or useful? What improvements would you suggest?

References

[ALG] Algorithm Analysis, Chapman University, 2023.