

# Lab 5 - Towers of Hanoi

Weichen Chai

Spring 2023

## Introduction

This paper seeks to answer the questions presented in "Towers of Hanoi" for ID1019. In this assignment, we will explore how the functional language Elixir can be utilized to implement a solution to this puzzle using recursive technique.

## Tasks

The puzzle we are looking to solve is the towers of Hanoi, which consists of pegs that is piled up from widest to slimiest from bottom to top. The disks can only be moved one at once and we want to stack all the disks onto the last rod. The rule is that no larger disks and be placed on smaller ones. Of course, the solution was proposed in the assignment to give us insight to the method in which to solve it, which was a recursive one. We can use the third stick as a temporary.

## Sequence

On to the actual code for this function to solve the tower of Hanoi puzzle, we will have three elements each representing a stick, then a tuple allowing us to see where the disk will move to. (ie, stick "a" to stick "b"). We then start implementing the function, starting with the most obvious, for size 0:

```
def hanoi(0, _, _, _) do [] end
```

Then for the recursive case, which is made up of three sequences we have

```
// from: stick 1  
// aux: stick 2  
// to: stick3  
def hanoi(n, from, aux, to) do
```

```

    hanoi(n-1, from, to, aux) ++
    // We move n-1 disks from stick 1 to 3 using 2
    [{:move, from, to}] ++
    // last disk to 3rd stick
    hanoi(n-1, aux, from, to)
    // from disk 2 to disk 3 using disk 1
end

```

Using a test giving in the assignment:

```
hanoi(3, :a, :b, :c)
```

We receive

```

[
  {:move, :a, :c},
  {:move, :a, :b},
  {:move, :c, :b},
  {:move, :a, :c},
  {:move, :b, :a},
  {:move, :b, :c},
  {:move, :a, :c}
]

```

Which is exactly what the steps should be when there are 3 disks. In order to further experiment with the algorithm, we can even go and test the towers of Hanoi puzzle with 4 disks with the result being

```

[
  {:move, :a, :b},
  {:move, :a, :c},
  {:move, :b, :c},
  {:move, :a, :b},
  {:move, :c, :a},
  {:move, :c, :b},
  {:move, :a, :b},
  {:move, :a, :c},
  {:move, :b, :c},
  {:move, :b, :a},
  {:move, :c, :a},
  {:move, :b, :c},
  {:move, :a, :b},
  {:move, :a, :c},
  {:move, :b, :c}
]

```

When there are 4 disks it consists of 15 moves, which has been shown here, after searching online for the correct steps to take, we realize that it is completely the same as the program outputs.

To answer the last question proposed in the assignment, we will test this algorithm with 10 disks. The answer should be 1023 according to the equation given for us to calculate the amount of steps it takes for the tower of Hanoi puzzle with disks of various sizes. However, the problem is that my compiler does not output the actual steps to the solution as it stops after around 33 moves. The recursive method has been explained in the code given above.