Program 2 - Tower of Hanoi

Michael Gould
CS 506 - Online Foundations of CS

March 12, 2018

First, write a recursive algortihm to solve the problem.

Using pseudo-code, the recursive algorithm to solving the towers of hanoi is as follows:

```
move(n, A, B, C)
   if(n = 1)
         move disk n from A to B
         end
   }
   else
         move(n-1, A, C, B)
         move disk n from A to B
         move(n-1, C, B, A)
   }
```

(See enclosed C program for specific implementation of the algorithm)

Write a recurrence equation for $f(n)$ and solve the recurrence. To get a better appreciation for this time complexity, tabulate $f(n)$ for the values of n from 1 to 20.

$$f(n) = \begin{cases} 1, & \text{n} = 1 \\ 2F_{n-1} + 1, & \text{n} > 1 \end{cases} \tag{1}$$

$f(n)$	Output
$F1$	1
$F2$	2(1) + 1 = 3
$F3$	2(3) + 1 = 7
$F4$	2(7) + 1 = 15
$F5$	2(15) + 1 = 31
$F6$	2(31) + 1 = 63
$F7$	2(63) + 1 = 127
$F8$	2(127) + 1 = 255
$F9$	2(255) + 1 = 511
$F10$	2(511) + 1 = 1,023
$F11$	2(1023) + 1 = 2,047
$F12$	2(2047) + 1 = 4,095
$F13$	2(4095) + 1 = 8,191
$F14$	2(8191) + 1 = 16,383
$F15$	2(16383) + 1 = 32,767
$F16$	2(32767) + 1 = 65,535
$F17$	2(65535) + 1 = 131,071
$F18$	2(131071) + 1 = 262,143
$F19$	2(262143) + 1 = 524,287
$F20$	2(524287) + 1 = 1,048,575

Solve the recurrance formula for f(n)
To solve this recurrane formula, we can observe a pattern from the above shown outputs for the first 20 instances of n. We notice that the growth is exponential and can assume that it also includes some constant as well, leaving us the guess:

$$f(n) = A2^n + b$$

Using the above guess as the hypothesis for our induction, we can start by using the base cases from the recurrance equation

$$f(1) = 1$$
$$f(1) = 2A + B$$

The first of the above equations is straight from the recurrance equation above, the latter is the solution form we are looking to achieve, therefore the goal is to find $2A + B = 1$
We begin by making our guess the hypothesis and assuming it is correct for some $n/geq1$:

$$f(n) = A2^n + B$$

Then we must also prove the solution is also correct for $n + 1$

$$f(n+1) = A2^{n+1} + B$$

To prove the conclusion, we can start with the recurrance equation with the input $n + 1$ and use the hypothesis to substitute in for $f(n)$. In addition to some distributive law as well we find

$$f(n+1) = 2f(n) + 1$$
$$f(n+1) = 2(A2^n + B) + 1$$
$$f(n+1) = A2^{n+1} + (2B + 1)$$
$$f(n+1) = A2^{n+1} + B$$

The leap in the last step is that of a term by term equaltiy, we will need this equality to help find the constant B which is can now be used in conjunction with our initial equation

$$2B + 1 = B$$
$$2A + B = 1$$

Using algebra we find that $A = 1$ and $B = -1$ and thus we have the constants to properly fill in the previous guess turned hypothesis

$$f(n) = A2^n + B$$

becomes

$$f(n) = 2^n + 1$$

Run your program for two values of n.

4 Disks	3 Disks
Tower of Hanoi Solver	Tower of Hanoi Solver
Input number of disks (max 6): 4	Input number of disks (max 6): 3
1 Move disc 1 from A to C	1 Move disc 1 from A to B
2 Move disk 2 from A to B	2 Move disk 2 from A to C
3 Move disc 1 from C to B	3 Move disc 1 from B to C
4 Move disk 3 from A to C	4 Move disk 3 from A to B
5 Move disc 1 from B to A	5 Move disc 1 from C to A
6 Move disk 2 from B to C	6 Move disk 2 from C to B
7 Move disc 1 from A to C	7 Move disc 1 from A to B
8 Move disk 4 from A to B	
9 Move disc 1 from C to B	
10 Move disk 2 from C to A	
11 Move disc 1 from B to A	
12 Move disk 3 from C to B	
13 Move disc 1 from A to C	
14 Move disk 2 from A to B	
15 Move disc 1 from C to B	