

## In-Class Problems Week 13, Mon.

### Problem 1.

The famous mathematician, Fibonacci, has decided to start a rabbit farm to fill up his time while he's not making new sequences to torment future college students. Fibonacci starts his farm on month zero (being a mathematician), and at the start of month one he receives his first pair of rabbits. Each pair of rabbits takes a month to mature, and after that breeds to produce one new pair of rabbits each month. Fibonacci decides that in order never to run out of rabbits or money, every time a batch of new rabbits is born, he'll sell a number of newborn pairs equal to the total number of pairs he had three months earlier. Fibonacci is convinced that this way he'll never run out of stock.

(a) Define the number,  $r_n$ , of pairs of rabbits Fibonacci has in month  $n$ , using a recurrence relation. That is, define  $r_n$  in terms of various  $r_i$  where  $i < n$ .

(b) Let  $R(x)$  be the generating function for rabbit pairs,

$$R(x) ::= r_0 + r_1x + r_2x^2 + \dots$$

Express  $R(x)$  as a quotient of polynomials.

(c) Find a partial fraction decomposition of the generating function  $R(x)$ .

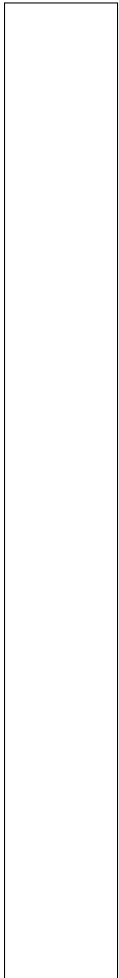
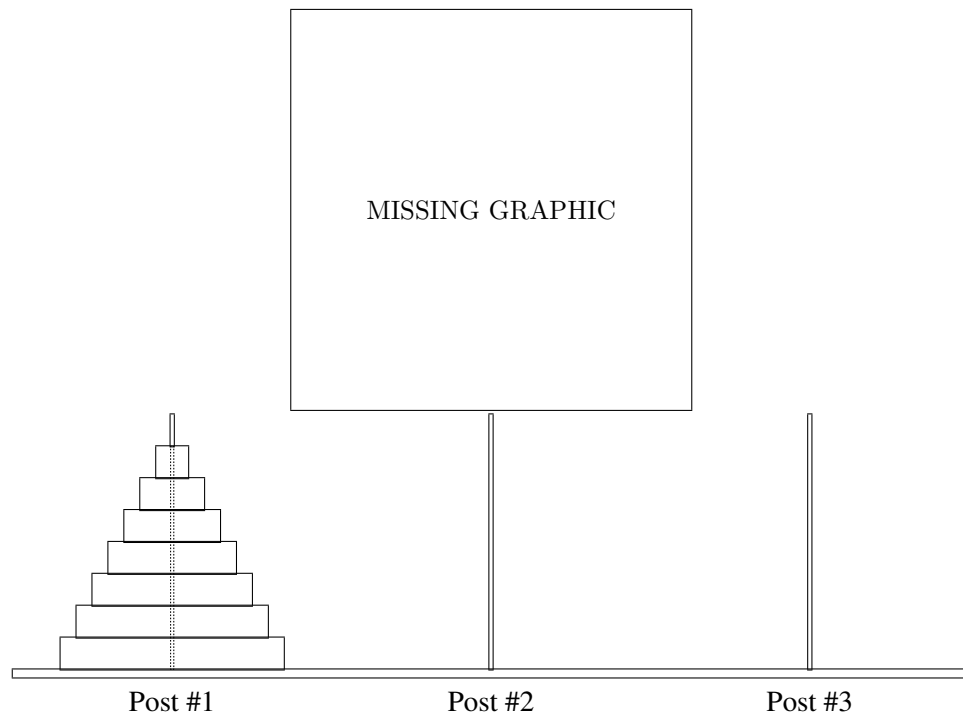
(d) Finally, use the partial fraction decomposition to come up with a closed form expression for the number of pairs of rabbits Fibonacci has on his farm on month  $n$ .

### Problem 2.

Less well-known than the Towers of Hanoi—but no less fascinating—are the Towers of Sheboygan. As in Hanoi, the puzzle in Sheboygan involves 3 posts and  $n$  disks of different sizes. Initially, all the disks are on post #1:

The objective is to transfer all  $n$  disks to post #2 via a sequence of moves. A move consists of removing the top disk from one post and dropping it onto another post with the restriction that a larger disk can never lie above a smaller disk. Furthermore, a local ordinance requires that *a disk can be moved only from a post to the next post on its right—or from post #3 to post #1*. Thus, for example, moving a disk directly from post #1 to post #3 is not permitted.

(a) One procedure that solves the Sheboygan puzzle is defined recursively: to move an initial stack of  $n$  disks to the next post, move the top stack of  $n - 1$  disks to the furthest post by moving it to the next post two times, then move the big,  $n$ th disk to the next post, and finally move the top stack another two times to land on top of the big disk. Let  $s_n$  be the number of moves that this procedure uses. Write a simple linear recurrence for  $s_n$ .



(b) Let  $S(x)$  be the generating function for the sequence  $\langle s_0, s_1, s_2, \dots \rangle$ . Carefully Show that

$$S(x) = \frac{x}{(1-x)(1-4x)}.$$

(c) Give a simple formula for  $s_n$ .



(d) A better (indeed optimal, but we won't prove this) procedure to solve the Towers of Sheboygan puzzle can be defined in terms of two mutually recursive procedures, procedure  $P_1(n)$  for moving a stack of  $n$  disks 1 pole forward, and  $P_2(n)$  for moving a stack of  $n$  disks 2 poles forward. This is trivial for  $n = 0$ . For  $n > 0$ , define:

$P_1(n)$ : Apply  $P_2(n-1)$  to move the top  $n-1$  disks two poles forward to the third pole. Then move the remaining big disk once to land on the second pole. Then apply  $P_2(n-1)$  again to move the stack of  $n-1$  disks two poles forward from the third pole to land on top of the big disk.

$P_2(n)$ : Apply  $P_2(n-1)$  to move the top  $n-1$  disks two poles forward to land on the third pole. Then move the remaining big disk to the second pole. Then apply  $P_1(n-1)$  to move the stack of  $n-1$  disks one pole forward to land on the first pole. Now move the big disk 1 pole forward again to land on the third pole. Finally, apply  $P_2(n-1)$  again to move the stack of  $n-1$  disks two poles forward to land on the big disk.

Let  $t_n$  be the number of moves needed to solve the Sheboygan puzzle using procedure  $P_1(n)$ . Show that

$$t_n = 2t_{n-1} + 2t_{n-2} + 3, \quad (1)$$

for  $n > 1$ .

*Hint:* Let  $s_n$  be the number of moves used by procedure  $P_2(n)$ . Express each of  $t_n$  and  $s_n$  as linear combinations of  $t_{n-1}$  and  $s_{n-1}$  and solve for  $t_n$ .

(e) Derive values  $a, b, c, \alpha, \beta$  such that

$$t_n = a\alpha^n + b\beta^n + c.$$

Conclude that  $t_n = o(s_n)$ .

## Appendix

Let  $[x^n]F(x)$  denote the coefficient of  $x^n$  in the power series for  $F(x)$ . Then,

$$[x^n] \left( \frac{1}{(1-\alpha x)^k} \right) = \binom{n+k-1}{k-1} \alpha^n. \quad (2)$$

### Partial Fractions

Here's a particular case of the Partial Fraction Rule that should be enough to illustrate the general Rule. Let

$$r(x) ::= \frac{p(x)}{(1-\alpha x)^2(1-\beta x)(1-\gamma x)^3}$$

where  $\alpha, \beta, \gamma$  are distinct complex numbers, and  $p(x)$  is a polynomial of degree less than the denominator, namely, less than 6. Then there are unique numbers  $a_1, a_2, b, c_1, c_2, c_3 \in \mathbb{C}$  such that

$$r(x) = \frac{a_1}{1-\alpha x} + \frac{a_2}{(1-\alpha x)^2} + \frac{b}{1-\beta x} + \frac{c_1}{1-\gamma x} + \frac{c_2}{(1-\gamma x)^2} + \frac{c_3}{(1-\gamma x)^3}$$

### Finding a Generating Function for Fibonacci Numbers

The Fibonacci numbers are defined by:

$$\begin{aligned} f_0 &::= 0 \\ f_1 &::= 1 \\ f_n &::= f_{n-1} + f_{n-2} \quad (\text{for } n \geq 2) \end{aligned}$$

Let  $F$  be the generating function for the Fibonacci numbers, that is,

$$F(x) ::= f_0 + f_1x + f_2x^2 + f_3x^3 + \dots$$

Now we have

$$\begin{array}{rcll} F(x) & = & f_0 & + \quad f_1x & + \quad f_2x^2 & + \quad f_3x^3 & + \dots \\ -x F(x) & = & & - \quad f_0x & - \quad f_1x^2 & - \quad f_2x^3 & - \dots \\ -x^2 F(x) & = & & & - \quad f_0x^2 & - \quad f_1x^3 & - \dots \\ \hline F(x)(1-x-x^2) & = & f_0 & + & (f_1-f_0)x & + & 0x^2 & + & 0x^3 & + \dots \\ & = & 0 & + & 1x & & & & & \end{array}$$

so

$$F(x) = \frac{x}{1-x-x^2}.$$