ArsDigita University
Month 2: Discrete Mathematics - Professor Shai Simonson

Problem Set 3 – Recursion and Induction

1. Solve the Chinese Rings Puzzle, also called the Patience Puzzle, (http://johnrausch.com/PuzzleWorld/patience.htm). This will prepare you for recursion, recurrence equations, proofs by induction and graph representations.

2. Consider the variation of the Towers of Hanoi Problem where you have four pegs instead of three. For simplicity you may assume that *n* is a power of two. Sloppy Joe designs this *solution:*

In order to move *n* disks from *From* to *To*, using *Using1* and *Using2*:
If *n* equals 1, then move a disk from *From* to *To,* otherwise do the three recursive steps:
Move *n/2* disks from *From* to *Using1*, using *To* and *Using2;*
Move *n/2* disks from *From* to *To*, using *Using1* and *Using2;*
Move *n/2* disks from *Using1* to *To*, using *From* and *Using2;*

   a. Explain why Sloppy Joe's solution does not work.
   b. Fruity Freddie suggests changing the second line:
      Move *n/2* disks from *From* to *To*, using *Using1* and *Using2;* to
      Move *n/2* disks from *From* to *To*, using *Using2* and *Using1;*
      Explain why the algorithm still does not work in general.
   c. Code the algorithms in Scheme and report what happens for $n = 4$, and $n = 8$.
   d. Even though neither solution above works, set up a recurrence equation for the number of moves it takes to run the algorithms.
   e. Solve this recurrence equation by repeated substitution.

3. Fix Joe and Freddie's failures.

   a. Construct a correct solution to the Towers of Hanoi problem with four pegs, that is faster than the standard solution with three pegs.
   b. Code your solution in Scheme and show the result for $n = 4$ and $n = 8$.
   c. Construct a recurrence equation for your solution and solve it.

4. For the 3-peg version, make a picture of the Towers of Hanoi graph for $n = 4$. Recall that this graph is composed of three copies of the Hanoi graph for 3 disks.

5. In the original Towers of Hanoi problem, alternately color the disks black and white as you go from the top to the bottom of the starting tower, including the base of the *From* tower. The color of the *To* base is the same as the color of the *From* tower, and different than the color of the *Using* tower.. Then in addition to the standard rules, add the constraint that only disks of different colors can be placed on top of one another, and a disk cannot be placed on a base with the same color.

    a. Show the path of the solution in the Towers of Hanoi graph for $n = 4$.
    b. Prove that aside from going backwards, there is no choice for each subsequent move.
    c. Prove by induction on the number of disks n, that the solution implied by this new rule, is identical to the standard recursive solution. Hint: First prove the following lemma by induction: For the Hanoi graph on n disks, let the node labeled with all i's be called i^n, $i \in \{1, 2, 3\}$. Then the rightmost occurrences of a and b, in the label of any node on the solution path from a^n to b^n, occur in odd numbered slots counting from right to left.

6. In the original Towers of Hanoi problem, add the constraint that no direct moves between the *From* peg to the *To* peg are allowed.

    a. Prove by induction, that following this new rule, will take you through every legal configuration of the game. Hint: Use the graph representation.
    b. Write a recurrence equation for the smallest number of moves it takes to solve the problem under the new constraint.
    c. Solve the equation.

7. Double-disk Hanoi has three pegs and n disks but there are two identical copies of each disk stacked on top of each other. The goal is the same as before, but we allow disks of the same size to reverse their order.

    a. Describe a method to solve this problem.
    b. Write and solve a recurrence equation for the number of moves implied by your solution.

8. Consider the standard Gray code sequence for binary numbers of length 4.

    a. Write down the standard Gray codes in order for binary numbers of length four.
    b. Draw a four-dimensional hypercube, labeling the vertices appropriately, and show that the Gray code ordering is a Hamiltonian Circuit through the Hypercube.

9. Prove by induction that the standard n-bit Gray code (defined inductively in terms of the $n-1$ bit Gray code) is a Hamiltonian Circuit through the n-dimensional hypercube.

10. Make a table of the 2^n elements of the Gray code for some *n*, by writing them in order starting from 0^n, one underneath the other.

    a. Conjecture a theorem regarding the patterns of the bits in the *i*th column of your chart. If you are having trouble writing down a theorem for the *i*th case, then you can try to write one down for a particular *i*.
    b. Prove your theorem.

11. Converting Gray codes. (The class handout will help with this).

    a. Write a Scheme program that takes a binary number *n* and outputs the *n*th element of a Gray code.
    b. Write a Scheme program that does the inverse of (a).

12. Consider the algorithm for computing a^n where *a* and *n* are integers. If $n = 0$, then return 1. If *n* is even then compute $a^{n/2}$ recursively, and square it. Otherwise, compute a^{n-1} recursively and multiply the result by *a*.

    a. How many multiplications does this method use when *n* is a power of two?
    b. How many multiplications when *n* is one less than a power of two?
    c. What exactly determines the number of multiplications for general *n*. Be as specific as possible.