# Homework 1 (Cpt S 223)

## Due Date: September 10, 2010

## Total points: 41

1. (3 points)

   Write a single recursive function to count the nodes in a simple binary tree $T$. The function should return the count of the number of nodes in $T$. Note that "nodes" include both internal nodes and leaf nodes. The tree node structure:

   ```
   class Node {
           Node *left;
           Node *right;
   }
   ```

   Assume that pointer $T$ initially points to the root of the binary tree. The function should have the following signature:

   *int CountNodes(Node *T)*

   Your answer can be in C++ syntax *or* in the form of a generic pseudocode.

2. (5 points)

   Evaluate the series: $\Sigma_{i=0}^{\infty} \frac{2\times i}{4^i} = ?$

   Show all the steps while deriving the answer.

   Hint: A technique for a similar problem is described in section 1.2.3 (page 4, Weiss).

3. (5 points)

   Solve the following mathematical recurrence:

   $$(1) \qquad \mathcal{T}(n) = \begin{cases} 1 & , n = 1 \\ 2\,\mathcal{T}(\frac{n}{2}) + n & , n > 1 \text{ and n is a power of 2} \end{cases}$$

   (Note: Your answer should be a closed form for $\mathcal{T}(n)$.)

4. (5 points)

   A 3-ary tree is one in which there are exactly 3 children for every internal node. Prove that the number of the nodes in a *complete 3-ary tree of level D* is: $\frac{3^{D+1}-1}{2}$.

   You can either use induction or provide a proof by construction — i.e., directly deriving the expression, whichever suits you. Recall that a D-level tree has exactly D levels under the root.

5. (5 points)

a) Using proof by contradiction, prove that:

$$\text{If } n^2 \text{ is odd then } n \text{ is odd.}$$

b) Using the above result, show the following:

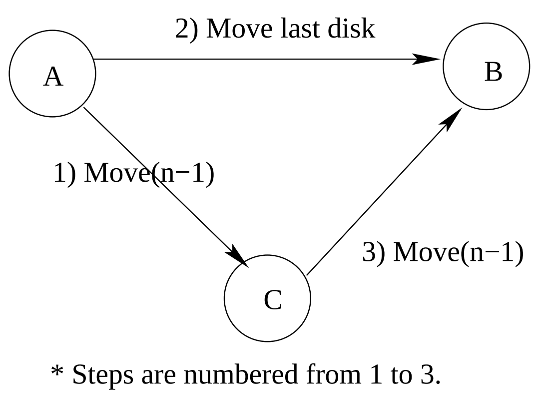
$$\text{If } n^2 \text{ is odd then } n^3 \text{ is odd.}$$

6. (8 points)

Rewrite the pseudocode presented in class (slide #37) for the Tower of Hanoi problem *without* tail recursion.

7. (10 points)

We discussed an algorithm to solve the Tower of Hanoi problem in class. The approach we discussed for optimally moving $n$ disks from peg A to peg B using peg C can be illustrated as shown below:




Therefore, number of moves in Move(n) is:

2 * Move(n−1) + 1

Now consider the following variant of the Tower of Hanoi problem. To the original problem, let us add a new condition: All moves now must be performed only in the *clockwise* direction — ie., $A \to B$, $B \to C$, or $C \to A$. So, for example, if I have to move a disk from A to C then that will involve at least two moves (one from A to B and then from B to C); but from C to A needs only one move. All the old conditions from the original version of the problem still continue to hold.

Under this new problem formulation, let $Q(n)$ denote the minimum number of moves required to transfer $n$ disks from $A \to B$ using C. Also, let $R(n)$ denote the minimum number of moves to transfer $n$ disks from $B \to A$ clockwise using C.

Show that:

(2)
$$\mathcal{Q}(n) = \begin{cases} 0 & , n = 0 \\ 2R(n-1)+1 & , n > 0 \end{cases}$$

(3)
$$\mathcal{R}(n) = \begin{cases} 0 & , n = 0 \\ Q(n)+Q(n-1)+1 & , n > 0 \end{cases}$$

Provide your answer in the form of an illustration similar to the figure shown above for the original version of the problem.