CS 325
Due: Friday, 1 Feb

# Homework #3

1. How many times is [ computation ] executed in the following nested loop program:

   for $i = 1$ to $2n - 1$ by step 2
       for $j = 1$ to $i$ by step 1
           [ computation ]

2. Use Euclid's algorithm to compute GCD(233, 144).

3. Let $f_n$ be the $n^{\text{th}}$ Fibonacci number and let $f_{n-1}$ be the $(n-1)^{\text{st}}$ Fibonacci number.

   (a) What is $gcd(f_n, f_{n-1})$?

   (b) Write down a difference equation for the number of recursive calls Euclid's algorithm makes in computing $GCD(f_n, f_{n-1})$.

   (c) Solve your difference equation from (b).

4. Show by induction that every connected graph with $n$ vertices has at least $n-1$ edges.

5. Use induction to show that

$$\sum_{i=0}^{n} i^3 \quad = \quad \left(\sum_{i=0}^{n} i\right)^2$$

6. Use induction to show that

$$\sum_{i=0}^{n} r^i = \frac{r^{n+1} - 1}{r - 1}, \qquad (r \neq 1)$$

   Interpret this formula as a result about addition in base $r$ when $r$ is a positive integer.

7. If the running time for an algorithm satisfies

$$T(n) = T(n-1) + T(n-2) + T(n-3)$$

   use induction to show that $T(n) = \Theta(\lambda_0^n)$ where $\lambda_0^3 = \lambda_0^2 + \lambda_0 + 1$. (HINT: For the base cases use the reasonable assumption that $T(3) > 0$, $T(2) > 0$, $T(1) > 0$. )

8. Show by induction that every function $\mathcal{F}$ such that $\mathcal{F} : \{0,1\}^n \mapsto \{0,1\}$ can be represented by composing only the boolean operations **AND**, **OR**, and **NOT**. That is, show how to expand $\mathcal{F}(x_1, x_2, \cdots, x_n)$ using only the variables $x_1, x_2, \cdots, x_n$ and the three allowed operators.

9. Use the following inductive hypothesis to show that the Buneman & Levy algorithm correctly solves the Towers of Hanoi problem.
   $B(K) =$ " After $2^K - 1$ moves B&L has moved the smallest $K$ disks one tower clockwise if $K$ is odd and one tower counterclockwise if $K$ is even. "

# The Towers of Hanoi Configurations

For the Towers of Hanoi problem, a Disk Array can represent $3^n$ possibilities, but only $2^n$ of these are actually used in the solution. Here, we want you to show that it is *easy* to determine whether or not a given Disk Array represents a configuration which is actually used in the solution.
INPUT: A disk array $D[1],\ D[2],\ \ldots\ D[n]$ with each $D[I]$ in $\{A,\ B\ C\}$.
QUESTION: Does the configuration defined by the array $D[1..n]$ ever occur in the minimal move solution to the Towers of Hanoi problem of moving $n$ disks from $A$ to $C$?

## Hints

THINK RECURSIVELY.
The algorithm $HANOI$ breaks the solution for an $n$ disk problem into TWO solutions for $n-1$ disk problems.
Can you tell which of these two subproblems you are solving by looking at the position of the $n^{\text{th}}$ disk?

## Problems

1. Design a *subtract-and-conquer* algorithm for the TofH configuration problem. Demonstrate by examples that your algorithm produces the correct answers.

2. Write and solve a difference equation for the running time of your algorithm.

3. Calculate the space used by your algorithm. Explain why you think that this is the minimum space required for any algorithm or why you think that some other algorithm might use less space.