CSE 20, Discrete Mathematics, Spring 199y
Homework 3,

Due Thursday Jan. 27, 1 PM Short problems: (2 points ea.)

- If $c_1 = 2$ and $c_{k+1} = 2c_k$, give an explicit formula for c_k.
- If $c_1 = 2$ and $c_{k+1} = (c_k)^2$, give an explicit formula for c_k.
- Epp. 4.1., pp. 192-194 , 23, 36, 44
- Epp, 8.1, pp. 438-440, 2, 6, 14, 22, 23

Full Proofs: (4 pts. ea.)

- Section 4.2, pp. 204-205: 2, 11
- Section 4.3, pp. 210-212: 12, 17
- Section 4.4., pp. 219-221: 2, 10
- Prove that for F_k the k'th Fibonnaci number, $F_k \leq 2^k$ for all $k \geq 0$. (Hint: use strong induction; see Epp 8.1. for the def. of Fibonacci number.)
- A bank maintains an array containing the current balance in each person's account, $A[1], ...A[n]$, where n is the number of accounts. It also keeps the sum of the accounts in a variable S. When the bank opens, all accounts are 0 and $S = 0$. After a customer i deposits D dollars, it updates $A[i] := A[i] + D$ and $S := S + D$. (View withdrawals as deposits with negative values). Prove, by induction on the number of deposits, that the invariant $S = \sum_{i=1}^{i=n} A[i]$ is maintained.

  (Hint: let S_t be the value of variable S after t deposits are made, and similarly for $A_t[i]$. Let D_t be the size of the t'th deposit, and let i_t be the number of the person making the t'th deposit. Prove $S_t = \sum_{i=1}^{i=n} A_t[i]$ by induction on t. Use the description of the banks updating algorithm to give expressions for S_{t+1} and $A_{t+1}[i]$ in terms of $S_t, A_t[i], D_{t+1}$. and i_{t+1}.)
- Consider the algorithm described by the following recursive "pseudo-code" program:

  1. Program: RPM {Russian Peasant Multiplication}
  2. Inputs: X,Y: Integer
  3. If $X = 0$ Return 0;
  4. If $X > 0$ and $X mod 2 = 0$ Return $2 * RPM(X/2, Y)$
  5. If $X > 0$ and $X mod 2 = 1$ Return $RPM(X - 1, Y) + Y$

Prove, by strong induction on X, that on any input X, Y with $X \geq 0$, $RPM(X,Y) = X * Y$.

Extra credit (2 points on final grade): What is the maximum number of recursive calls RPM might make on X? Prove your answer is correct.

- Consider the algorithm described by the following "pseudo-code" program:

```
Program:1B
Input:x
begin;
y:= 0;
While x > 0 do:
beginwhile;
   If x MOD 2 =1 THEN x:= x-1; y := y+1;
                 ELSE x:= x/2;
endwhile;
Output y;
end;
```

  Prove that for any positive integer n, when given input $x = n > 0$, the while loop is executed at most $2(\log_2 n) + 1$ times.

  Hint: One approach: use a proof by strong induction on the value of x. Second approach:use a normal induction proof on t to show that if the program has not halted before $2t$ loops, then after the $2t$'th loop, the value of x is at most $n/2^t$. Then use this to prove the time bound.

- Extra Credit: (2 pts. on final grade) What is the function that the above program computes? Prove your answer correct.

- Extra Credit: (3 points on final grade) There are R red balls and B black balls in a bin. You repeat the following process: Remove two balls arbitrarily from the bin. If the balls have the same color, put a new red ball in the bin, otherwise, put a black ball back in the bin. The process ends when only one ball is left in the bin. Prove that the color of the ball left at the end only depends on R and B, not on the order in which balls are selected.