Math 55: Discrete Mathematics, Fall 2008
Homework 6 Solutions

4.3: 6(b) The function is well-defined and given by the formula

$$f(n) = \begin{cases} 2^{n/3} & \text{for } n \equiv 0 \pmod 3 \\ 0 & \text{for } n \equiv 1 \pmod 3 \\ 2^{(n+1)/3} & \text{for } n \equiv 2 \pmod 3 \end{cases}$$

To prove it, observe that this gives the correct initial values $f(0) = 1$, $f(1) = 0$, $f(2) = 2$, and that for $n \geq 3$ the above formula satisfies the recurrence $f(n) = 2f(n-3)$.

* 12. We prove that $f_1^2 + \cdots + f_n^2 = f_n f_{n+1}$ by induction on n. Basis step $n = 1$ is true, as both sides are equal to 1. For $n > 1$, assume by induction that $f_1^2 + \cdots + f_{n-1}^2 = f_{n-1}f_n$. Adding f_n^2 to both sides gives $f_1^2 + \cdots + f_n^2 = f_{n-1}f_n + f_n^2 = f_n(f_{n-1} + f_n) = f_n f_{n+1}$.

30. Let's denote the number of occurences of 01 in a bit string w by $b(w)$ and the number of occurences of 10 by $a(w)$. We are to prove that $b(w) \leq a(w) + 1$ for any given bit string w. We may assume by strong induction that the inequality holds for all shorter bit strings.

If w is empty or has length 1, then $a(w) = b(w) = 0$, so we can assume w has length at least 2. If w ends with 0, say $w = v0$, then $b(w) = b(v) \leq a(v) + 1$ by induction, and clealy $a(v) \leq a(w)$, so $b(w) \leq a(w) + 1$. If w ends with 11, say $w = v11$, the same reasoning applies.

If w ends with 1 and all the other letters are 0, then $b(w) = 1$, $a(w) = 0$, so the result holds. Otherwise, $w = u100\ldots01$, with at least one 0 between the last two 1's. Then $b(w) = b(u1) + 1$, and $a(w) = a(u1) + 1$. The result follows since $b(u1) \leq a(u1) + 1$ by induction.

Ch. 4 Suppl. Ex. 18. We have $3|f_n$ iff $n \equiv 0 \pmod 4$. Here are two possible proofs.

Method 1. Prove the stronger statement that $f(n) \pmod 3$ is a function of $n \pmod 8$, given by the following table:

$n \pmod 8$	0	1	2	3	4	5	6	7
$f(n) \pmod 3$	0	1	1	2	0	2	2	1

To prove this by strong induction, observe that it is correct for $n = 0, 1$. For $n > 1$, we have $f_n \equiv f_{n-1} + f_{n-2} \pmod 3$, so we just have to observe that the entry for each $n \pmod 8$ in the table is the sum modulo 3 of the entries for $n - 1 \pmod 8$ and $n - 2 \pmod 8$, which is true. (Note that this includes the fact that the entry for $n \equiv 0 \pmod 8$ is the sum of those for 6 and 7, and the entry for $n \equiv 1 \pmod 8$ is the sum of those for 7 and 0.)

Method 2. Observe that the statement is true for $n = 0, 1, 2, 3$, and prove it by induction for $n > 3$ by showing that for all $n \geq 0$, we have $3|f_{n+4}$ if and only if $3|f_n$. Using the defining recurrence repeatedly, we get $f_{n+4} = f_{n+3} + f_{n+2} = 2f_{n+2} + f_{n+1} = 3f_{n+1} + 2f_n$. The last expression is congruent to $-f_n \pmod 3$, so $f_{n+4} \equiv 0 \pmod 3$ if and only if $f_n \equiv 0 \pmod 3$.

* [5 pts each part] 4.4: 48(a) We can merge a list of one element (x) with a list of four elements ($y_1 < y_2 < y_3 < y_4$) using 3 comparisions (one less than Algorithm 10) by first

comparing x with y_2, then if $x < y_2$ compare with y_1, otherwise compare with y_3 and y_4. There are five possible outcomes for the merged list, as x may end up at the beginning or after any of the four y_i's. Hence no merging algorithm using at most 2 comparisons is possible, since this could only distinguish $2^2 = 4$ outcomes.

(b) Algorithm 10 can merge a list of 2 and a list of 4 using 5 comparisons. We will show that it cannot be done with only 4 comparisons. This is a bit tricky as the number of outcomes for merging $(x_1 < x_2)$ with $(y_1 < y_2 < y_3 < y_4)$ is 15, which in principle might be distinguished with 4 comparisons. However, you can check that no matter what you do for the first comparison, it always splits the 15 possible outcomes into either 5 cases versus 10, or 6 versus 9, or 3 vs 12, or 1 vs 14. So one of two the possible results of the first comparison will require 4 addisional comparisons to distingish all remaining cases.

5.1: 16. $26^4 - 25^4$.

20(a) 142, (b) 130, (c) 12, (d) 220, (e) 208, (f) 780

28. $26^3 10^3 + 26^4 10^2$

38. $2^{100} - 101$

42. $2^5 + 2^4 - 2^2$

54. There are 70 possible ways the World Series can occur.

58. A truth table for a propositional function P of n variables has 2^n rows, one for each combination of truth values T or F for each of the n variables. In each row we must specify a truth value for P, giving 2^{2^n} possible propositional functions.

* (A) If our tree is just a root, $T = \{r\}$, then $n(T) = l(T) = 1$ and $h(T) = 0$, so the inequality $n(T) \le l(T)h(T) + 1$ holds.

Otherwise, our tree is $T = (r, T_1, \ldots, T_k)$ and we can assume that each of the smaller trees $T_1, \ldots, T_k$ satisfies the inequality. Then

$$\begin{aligned} n(T) &= 1 + n(T_1) + \cdots + n(T_k) \\ &\le k + 1 + l(T_1)h(T_1) + \cdots + l(T_k)h(T_k) \quad \text{by induction} \\ &\le k + 1 + (h(T) - 1)(l(T_1) + \cdots + l(T_k)) \quad \text{since } h(T_i) \le h(T) - 1 \text{ for all } i \\ &= k + 1 + (h(T) - 1)l(T) \\ &\le h(T)l(T) + 1. \end{aligned}$$

In the last step we used the inequality $l(T) \ge k$, which holds because $l(T) = l(T_1) + \cdots + l(T_k)$ and each $l(T_i) > 0$, since the trees T_i are non-empty.