CS 173: Discrete Mathematical Structures, Spring 2008
Homework 7

Due *in class* on Thursday, March 13, 2008

1. (a) Suppose that you need to construct a tiling of a strip that is $1 \times n$ in dimension. You have two types of tile available to you: there are tiles of size 1×1 and there are tiles of size 1×2. Find a recursive definition for the number of tilings that are possible, and represent your final solution in terms of F_n.

   (b) Suppose that you find on your CS 173 instructor's desk a toy with three wooden pegs, one of which has 8 disks neatly stacked on it. A card near the toy tells you the rules of playing:

   - Your goal is to move all of the disks from the first peg to one of the other two pegs.
   - Only one disk may be moved at a time.
   - No disk may ever be placed on a smaller disk.

   Let R_n be the minimum number of steps required to move n disks on a peg of such a toy to another peg. Impress your instructor by giving a recurrence for R_n, and justify your answer.

2. Recall that the Fibonacci numbers F_n for $n \geq 0$ are defined as follows: $F_0 = 0, F_1 = 1$, and for $n \geq 2$, $F_n = F_{n-1} + F_{n-2}$. As you recall from lecture, everyone knew immediately that the closed form of this recurrence was $F_n = \frac{1}{\sqrt{5}}(\phi^n - \hat{\phi}^n)$, where $\phi = \frac{1+\sqrt{5}}{2}$ and $\hat{\phi} = \frac{1-\sqrt{5}}{2}$. Prove this using induction on n.

3. Solve the following recurrences. State tight asymptotic bounds for each function in the form $\Theta(f(n))$ for some recognizable function $f(n)$. Show proofs (either by induction or using summations and recursion trees) that your bounds are correct. However, you do not need to prove that your big-O bounds are formally correct; for example, if you get a bound of $\frac{n}{2}\log n + 12$, you may just say that it is $O(n \log n)$.

   (a) $A(1) = 1$, and for $n \geq 2$ with n a power of 2, $A(n) = 4A(n/2) + n^2$
   (b) $B(1) = 1$, and for $n \geq 2$ with n a power of 4, $B(n) = 3B(n/6) + n$
   (c) $C(1) = 1$ and for $n \geq 2$ with n a power of 3, $C(n) = 4C(n/3) + n$
   (d) $D(1) = 1$, and for $n \geq 2$ with n a power of 3, $D(n) = D(n/3) + \log_3 n$
   (e) $E(1) = 2$, $E(2) = 3$, and for $n \geq 3$, $E(n) = E(n-2) + n + 12$

4. Let $U_0 = \emptyset$, and define for each $n \geq 1$ $U_n = \{1, 2, 3, \ldots, n\}$. Let $R_n(k)$ be the number of one-one functions from U_k into U_n. For example, $R_n(2) = n^2 - n$. Give a recursive definition for $R_n(k)$, and briefly explain your answer. *(Hint: For the base case, remember the adage: "There's one way to do nothing.")*

5. Let X_n be the number of different ways of parenthesizing the product of n values. For example, $X(1) = X(2) = 1$, $X(3) = 2$ (they are $(xx)x$ and $x(xx)$), and $X(4) = 5$; they are $x((xx)x)$, $x(x(xx))$, $(xx)(xx)$, $((xx)x)x$, and $(x(xx))x$. Prove that if $n \leq 2$, then $X(n) = 1$, and otherwise
$$X(n) = \sum_{k=1}^{n-1} X(k) \cdot X(n-k)$$

6. **[Honors]** An *elevated Schröder path of order* n is a path from $(0,0)$ to $(2n,0)$ in the first quadrant using steps in $\{(1,1),(2,0),(1,-1)\}$. In other words, for each move in an elevated Schröder path, we can either:

   - Move right one space, and up one
   - Move right two spaces, and not move up
   - Move right one space, and down one

   An *uprun* is a maximal segment of upward steps (a segment of upward steps is "maximal" if it is not contained in another longer segment of upward steps).

   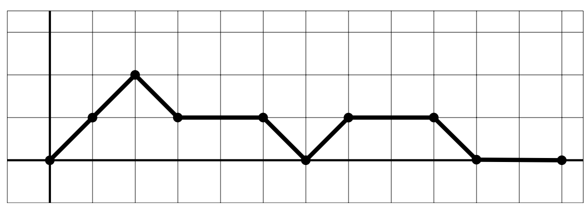

   Figure 1. An elevated Schröder path of order 6 with two upruns.

   (a) How many elevated Schröder paths of order n have no upruns?

   (b) How many elevated Schröder paths of order n have n upruns?

   (c) Let $A(n)$ be the number of elevated Schröder paths of order n having one uprun. Find a recurrence relation for $A(n)$ and justify your answer. *(Hint: What's the last step the path takes?)*