Mathematics for Computer Science
Spring 2019
Due: 23:59, April 8, 2019

Homework Set 6

Announcement: The *Midterm Exam* will take place in class on May 7, 2019. You can use any class materials (textbooks, homework set solutions, notes, etc), but do not use computers or the internet during the Exam.

Do the following special problems:

Special Problem 1 (counted as 3 exercises) Show that for some fixed constants $c, c' > 0$, the randomized routing algorithm discussed in class has the following performance for Phase 2:

$$\Pr\{T > cn\} \leq 2^{-c'n}.$$

Special Problem 2 (counted as 2 exercises) Solve each of the following recurrence relations:
(a) $a_0 = 1, a_1 = 2, a_n = 4a_{n-1} - 3a_{n-2} + 3n + 1$ for all $n \geq 2$.
(b) $a_0 = 1, a_n = \frac{a_{n-1}}{1+3a_{n-1}}$ for $n \geq 1$.

Special Problem 3 (counted as 4 exercises) Consider a sequence of $2n$ people in a line at a cashier. Suppose n of the people pay 1 yuan each and n of the people get 1 yuan each. A *paying pattern* is a binary sequence $\sigma = a_1 a_2 \cdots a_{2n}$ with exactly n 1's and n 0's; the interpretation is that $a_j = 1$ if person j pays 1 yuan, and $a_j = 0$ otherwise. Note that there are exactly $\binom{2n}{n}$ paying patterns. Let b_n denote the number of paying patterns in which the cashier never goes in debt (i.e., at every stage at least as many people have paid in 1 yuan as were paid out 1 yuan).
(a) Derive a recurrence equation for b_n, and find an explicit expression for the generating function $\sum_{n \geq 1} b_n x^n$. Determine a closed-form solution for b_n.
(b) Use an alternative way (a *combinatorial proof*) to establish a closed-form solution $b_n = \binom{2n}{n} - \binom{2n}{n+1}$. (**Hint:** Show a one-to-one correspondence between paying patterns where at some stage the cashier goes at least 1 yuan in debt and all binary sequences of length $2n$ with exactly $n+1$ 1's.)