BU CLA CS 530: Analysis of Algorithms
Spring 1999

Selected Solutions for Problem Set 7

Problem 1.
(a) (Exercise 6.2-3, page 109, in [CLR])
Any 3 cards are equally probable to be chosen. For any set of 3 chosen cards, any of the 6 orderings of the 3 cards are equally probable. For example, if cards with numbers 1, 2, and 3, are selected, then the possible orderings are:

1	2	3
1	3	2
2	1	3
2	3	1
3	1	2
3	2	1

The probability that the 3 chosen cards are in increasing order is therefore 1/6.
(b) Suppose $i, j, k \in \{1, 2, \ldots, 2n\}$ where $n \geqslant 1$ and i, j and k are distinct. There are exactly two ways in which $i + j + k$ add up to an even number:

(X) all 3 integers in $\{i, j, k\}$ are even,

(Y) exactly 1 integer in $\{i, j, k\}$ is even.

The number of 3-subsets satisfying (X) is $\binom{n}{3} = n \cdot (n-1) \cdot (n-2)/6$. The number of 3-subsets satisfying (Y) is $n \cdot \binom{n}{2} = n \cdot n \cdot (n-1)/2$. Hence, the number of 3-subsets satisfying (X) or (Y) is $\binom{n}{3} + n \cdot \binom{n}{2} = n \cdot (n-1) \cdot (4n-2)/6$. The total number of 3-subsets of $\{1, 2, \ldots, 2n\}$ is $\binom{2n}{3} = 2n \cdot (2n-1) \cdot (2n-2)/6$. Hence, the probability of selecting 3 numbers i, j and k from $\{1, 2, \ldots, 2n\}$ that add up to an even number is:

$$\frac{n \cdot (n-1) \cdot (4n-2)/6}{2n \cdot (2n-1) \cdot (2n-2)/6} \quad = \quad \frac{1}{2} \cdot \frac{n-1}{2n-1} \cdot \frac{2(2n-1)}{2(n-1)} \quad = \quad \frac{1}{2}$$

Problem 2.
Letting $g(n) = f(n)/n$ in the given recurrence:

$$f(n) = \sqrt{n} f(\sqrt{n}) + c \cdot n$$

we obtain the recurrence:

$$g(n) = g(\sqrt{n}) + c$$

Letting $h(n) = g(2^{2^n})$, the preceding recurrence is transformed into:

$$h(n) = h(n-1) + c$$

The solution of this last recurrence is $h(n) = \mathcal{O}(n)$. (The simplest way is to solve it by iteration.) This implies that $g(n) = \mathcal{O}(\log\log n)$, and therefore $f(n) = \mathcal{O}(n \log\log n)$.

Problem 3.

(a) If n is a power of 2, then the upper part of the recurrence will always be used. In this case, we can substitute k for n, and obtain the following recurrence for the function $T'(k) = T(2^k)$ for all $k \geq 0$:

$$T'(k) \;=\; T'(k-1) + 1$$

with boundary condition $T'(0) = 1$. It is easy to check that $T'(k) = k + 1$ is the solution of this recurrence.

(b) Consider $n = 2^k - 1$ for arbitrary $k \geq 1$. If we use the bottom part of the recurrence, then $T(n)$ is replaced by $T((n-1)/2)$ and $(n-1)/2 = 2^{k-1} - 1$. Thus, if we start with $n = 2^k - 1$ then we will always use the bottom part of the recurrence. Defining now the function $T'(k) = T(2^k - 1)$ for all $k \geq 1$, we obtain the recurrence:

$$T'(k) \;=\; 2 \cdot T'(k-1) + 1$$

with boundary condition $T'(1) = 1$. It is easy to check that $T'(k) = 2^{k-1}$ is the solution of this recurrence.

(c) We can restrict attention to powers of 2 if the solution of the recurrence is a monotonically increasing function. (This is the case for most algorithms encountered in practice, whose running times are indeed monotonically increasing functions.) In such a case, if $T(n)$ is the solution of the recurrence for n ranging over powers of 2, then $T(m)$ for $n < m < 2n$ is no more than a constant times $T(n)$.

Problem 4. (Exercise 27.3-3, page 604 in [CLR])

Let G_f be the residual graph of the flow network G' relative to some flow f. By definition, an augmenting path is a simple path from s to t in the residual graph G_f. Since G has no edges between vertices in L and no edges between vertices in R, neither does the flow network G' and hence neither does G_f. Also, the only edges involving s or t connect s to L and R to t. Note that although edges in G' can go only from L to R, edges in G_f can also go from R to L. Hence, any augmenting path must be of the form:

$$s \longrightarrow L \longrightarrow R \longrightarrow L \longrightarrow \;\cdots\; \longrightarrow L \longrightarrow R \longrightarrow t$$

crossing back and forth between L and R at most as many times as it can do so without using a vertex twice (because the path is simple). It contains s, t, and equal numbers of distinct vertices from L and R, for a maximum of at most $2 + 2 \cdot \min(|L|, |R|)$ vertices. The length of an augmenting path is therefore at most $1 + 2 \cdot \min(|L|, |R|)$.

Assaf Kfoury
Created: 99.03.04
Modified: 99.03.10