Discrete Structures EPFL, Fall 2013

Solution to Problem Set 9

Date: 15.11.2013 Not graded

Problem 1.

(a) Let a_n be the answer to the problem and $n = 2^k$. We have the following recursion : $a_1 = 1$, $a_{2^k} = 2^k + a_{2^{k-1}}$, which follows immediately from the structure of the algorithm.
Claim: For all $k \geq 0$ we have that $a_{2^k} = 2^{k+1} - 1$.
Proof by induction.
Base step: $a_{2^0} = 2^1 - 1 = 1$. [And indeed, if $n = 1$, we print the sentence once.]
Induction step: Suppose that for $k \geq 0$ it is true that $a_{2^k} = 2^{k+1} - 1$. Then, using the recursion, we obtain $a_{2^{k+1}} = 2^{k+1} + 2^{k+1} - 1 = 2^{k+2} - 1$.

As a result, the phrase is printed $2^{k+1} - 1 = 2n - 1$ times.

(b) The main observation here is that after a "west" step, a "south" step necessarily follows, and also a "south" step is followed by a "west" step if y is still > 0.
Also the assumption "$x > y$" is preserved during the recursive call: in case $x + y$ is even, x and y have the same parity, so $x - y \geq 2$ and the condition "$x > y$" is preserved after x is decremented. In case $x + y$ is odd, y is decremented, and, therefore, the condition still holds.
After trying out some values for x and y, we can make the following conjecture. If $f(x, y)$ is the number of times the algorithm prints a sentence (any) when presented with parameters x and y, then

$$f(x,y) = \begin{cases} 2y, & \text{if } x+y \text{ is even} \\ 2y-1, & \text{if } x+y \text{ is odd, and } y \neq 0 \\ 0, & \text{if } x+y \text{ is odd, and } y = 0 \end{cases}$$

Proof by induction (on y). The proposition that we want to prove is $P(y)$, meaning "For all x greater than y, the number of times the algorithm prints a sentence is $f(x, y)$", which depends only on y.

Base step. If $y = 0$, then we print 0 times a sentence, since we never enter into the **if** construct, regardless of the parity of $x + y$.

Induction step. Assume that the proposition $P(y - 1)$ is true for some $y \geq 1$. We will prove that $P(y)$ is also true.

- Let x be such that $x + y$ is even. After two recursive calls, the parameters become $(x', y') = (x-1, y-1)$ and clearly $x' + y'$ is even. By induction hypothesis, $f(x-1, y-1) = 2(y-1)$. Then $f(x, y) = 2 + f(x-1, y-1) = 2y$.
- Let x be such that $x + y$ is odd. After one recursive call the parameters become $(x', y') = (x, y-1)$ and clearly $x' + y'$ is even. In this case, we print a sentence $1 + f(x, y-1) = 1 + 2(y-1) = 2y - 1$ times, which is exactly $f(x, y)$.

Problem 2.

(a)

Algorithm 1 Mult($a \in \mathbb{R}, n \in \mathbb{N}_{\geq 1}$)

if $n = 1$ **then return** a
else return $a +$ Mult($a, n - 1$)

(b)

Algorithm 2 Power($n \in \mathbb{N}_{\geq 0}$)

if $n = 0$ **then return** 3
else return Mult(Power($n - 1$), Power($n - 1$))

(c) The complexity of the algorithm Mult is $O(n)$.

Algorithm 3 Mult-opt($a \in \mathbb{R}, n \in \mathbb{N}_{\geq 1}$)

if $n = 1$ **then return** a
$k \leftarrow \lfloor \frac{n}{2} \rfloor$
$b \leftarrow$ Mult-opt(k,a)
if n is even **then return** $b + b$
else return $a + b + b$

The complexity of the algorithm Mult-opt is $O(\log n)$.

Problem 3.

(a) The first 3 digits are fixed, while the remaining 9 are free. Hence, this is equivalent to counting the bit strings of length 9, of which there are exactly 2^9.

(b) We fix the first 2 and the last 2 digits, while the remaining 8 are free. Therefore, there are 2^8 possible bit strings.

(c) The number of bit strings which begin with 11 is 2^{10}, the number of bit strings which end with 10 is 2^{10} and the number of bit strings which begin with 11 and end with 10 is 2^8. By inclusion-exlusion principle the number of bit strings which begin with 11 or end with 10 is

$$2^{10} + 2^{10} - 2^8 = 2^{11} - 2^8 = 2^8 \cdot 7$$

(d) Counting the bit strings with exactly 4 1's is equivalent to counting the ways in which one can choose the positions of these four 1's. These are the combinations of 4 elements from a class of 12 without replacement. Therefore the solution is $\binom{12}{4}$.

(e) (*Bonus.*) Let A be the set of bit strings of length 12 which have exactly four 1's such that none of these 1's are adjacent to each other. Let B be the set of bit strings of length 9 with exactly four 1's.
Consider a string $a \in A$: it has four 1's which are not adjacent to each other. Hence, each of the first three 1's has to be followed by a 0. Consider the function $f : A \to B$ which maps the string $a \in A$ into the string $b \in B$ such that b is obtained removing the three 0's immediately after each of the first three 1's of a. It is easy to check that the function f is a bijection. Consequently, so $|A| = |B|$. By reasons similar to those of point (d), $|B| = \binom{9}{4}$ and, as a result, the solution is $\binom{9}{4}$.

Problem 4.

(a1) Clearly, $E_i \subset E$ for any $i \in \{0, 1, \cdots, n\}$. Therefore $E_0 \cup E_1 \cup \cdots \cup E_n \subset E$. Also, $E \subset E_0 \cup E_1 \cup \cdots \cup E_n$, because every string has a number of 1's which is an integer between 0 and n. Consequently, $E = E_0 \cup E_1 \cup \cdots \cup E_n$.

(a2) Counting the bit strings with exactly i 1's is equivalent to counting the ways in which one can choose the positions of these i 1's. These are the combinations of i elements from a class of n without replacement. Therefore, $E_i = \binom{n}{i}$.

(a3) $|E_i \cap E_j| = 0$ for $i \neq j$, because a binary string cannot have simultaneously i and j 1's with $i \neq j$.

(a4) The number of binary strings of length n, i.e., the cardinality of E, is equal to 2^n. In addition, $|E| = \sum_{i=0}^{n} |E_i|$, because the sets E_i for $i \in \{0, 1, \cdots, n\}$ are pairwise disjoint by point (a3). Using this last equation and the result of point (a2), we obtain that

$$2^n = |E| = \sum_{i=0}^{n} |E_i| = \sum_{i=0}^{n} \binom{n}{i}.$$

(b1) Recall that $\binom{m}{n} = \frac{m!}{n!(m-n)!}$. Then, for any $m \geq 1$ and any $1 \leq n \leq m-1$,

$$\binom{m-1}{n} + \binom{m-1}{n-1} = \frac{(m-1)!}{n!(m-1-n)!} + \frac{(m-1)!}{(n-1)!(m-n)!}$$
$$= \frac{(m-1)!(m-n+n)}{n!(m-n)!} = \frac{m!}{n!(m-n)!} = \binom{m}{n}.$$

(b2) *Base step.* For $n = 0$, $\sum_{i=0}^{n} \binom{n}{i} = 1$ and $2^n = 1$.

Induction step. Assume that $\sum_{i=0}^{n} \binom{n}{i} = 2^n$. Then, using the identity that has been proved in point (b2) and the induction hypothesis, we obtain that

$$\sum_{i=0}^{n+1} \binom{n+1}{i} = \binom{n+1}{0} + \binom{n+1}{n+1} + \sum_{i=0}^{n-1} \binom{n+1}{i+1} = 2 + \sum_{i=0}^{n-1} \binom{n}{i} + \sum_{i=0}^{n-1} \binom{n}{i+1}$$
$$= \binom{n}{n} + \binom{n}{0} + \sum_{i=0}^{n-1} \binom{n}{i} + \sum_{i=1}^{n} \binom{n}{i} = 2\sum_{i=0}^{n} \binom{n}{i} = 2 \cdot 2^n = 2^{n+1}.$$