Discrete Structures — EPFL, Fall 2013

Problem Set 9

Date: 15.11.2013 — Not graded

Problem 1. Consider the following algorithms :

Algorithm 1 LoopyRecursion

Require: n: positive integer
for $i = 1, \ldots, n$ **do**
  print "Counting is fun"
if $n > 1$ **then**
  LoopyRecursion$\left(\lfloor \frac{n}{2} \rfloor\right)$

Algorithm 2 TwoParameterRecursion

Require: x, y: positive integers
if $x > 0$ and $y > 0$ **then**
  if $x + y$ is even **then**
    print "Go West!"
    TwoParameterRecursion$(x - 1, y)$
  else
    print "Go South!"
    TwoParameterRecursion$(x, y - 1)$

(a) How many times is the phrase "Counting is fun" printed in **Algorithm 1**? To make your life easier, only look at the case where n is a power of 2.

(b) In **Algorithm 2**, assume $y < x$. How many times do you print either "Go West!" or "Go South!"?
Hint: try to draw a picture...

Problem 2. Suppose that the only operation your computer is able to perform on real numbers is the addition (and suppose that on integers it works like a regular machine).

(a) Describe a recursive algorithm to evaluate $n \cdot a$, where $n \in \mathbb{N}_{n\geq 1}$ and a is a real number.

(b) Describe a recursive algorithm to evaluate $3^{(2^n)}$, where $n \in \mathbb{N}_{n\geq 0}$. You may call the algorithm developed in point a).

(c) Find the best big-O approximation of the number of additions your algorithm performs in point (a). Optimize the algorithm in order to reduce this number to $O(\log n)$. If your algorithm already achieves $O(\log n)$, then... congratulations :)
Hint: think of the algorithm used for modular exponentiation...

Problem 3. Consider all bit strings of length 12.

(a) How many begin with 110?

(b) How many begin with 11 and end with 10?

(c) How many begin with 11 or end with 10?

(d) How many have exactly four 1's?

(e) (*Bonus.*) How many have exactly four 1's, none of which are adjacent to each other?

Problem 4. The aim of this exercise is to prove that for all $n \geq 0$,

$$\sum_{i=0}^{n} \binom{n}{i} = 2^n. \tag{1}$$

(a1) First of all, let us focus on a purely combinatorial proof of the claim. Denote by E the set of binary strings of length n and by E_i the set of strings which contain exactly i 1's for $i \in \{0, 1, \cdots, n\}$. Convince yourself that $E = E_0 \cup E_1 \cup \cdots \cup E_n$.

(a2) Compute $|E_i|$ for all $i \in \{0, 1, \cdots, n\}$.

(a3) Compute $|E_i \cap E_j|$ for all $i, j \in \{0, 1, \cdots, n\}$ with $i \neq j$.

(a4) Conclude evaluating $|E|$ directly and with the inclusion-exclusion principle.

(b1) Then, let us focus on an algebraical proof. Prove that for any $m \geq 1$ and any $1 \leq n \leq m-1$,

$$\binom{m}{n} = \binom{m-1}{n} + \binom{m-1}{n-1}. \tag{2}$$

(b2) Use formula (2) to prove (1) by the Principle of Mathematical Induction.