Eidgenössische Technische Hochschule Zürich
Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Swiss Federal Institute of Technology Zurich

Institut für Theoretische Informatik
Dr. B. Gärtner, Prof. Dr. J. Hromkovič

24. November 2009

Informatik für Mathematiker und Physiker — Serie 10 — HS 09

URL: http://www.ti.inf.ethz.ch/ew/courses/Info1_09/

Skript-Aufgabe 113 (5 Punkte)

a) Write and test a C++ function that computes binomial coefficients $\binom{n}{k}, n, k \in \mathbb{N}$. These may be defined in various equivalent ways. For example,

$$\binom{n}{k} := \frac{n!}{k!(n-k)!},$$

or

$$\binom{n}{k} := \begin{cases} 0, & \text{if } n < k \\ 1, & \text{if } n = k \text{ or } k = 0 \\ \binom{n-1}{k} + \binom{n-1}{k-1}, & \text{if } n > k, k > 0 \end{cases},$$

or

$$\binom{n}{k} := \begin{cases} 0, & \text{if } n < k \\ 1, & \text{if } n \geq k, k = 0 \\ \frac{n}{k}\binom{n-1}{k-1} & \text{if } n \geq k, k > 0 \end{cases}$$

b) Which of the three variants is best suited for the implementation, and why? Argue theoretically, but underpin your arguments by comparing at least two different implementations of the function.

Skript-Aufgabe 114 (6 Punkte)

In how many ways can you own CHF 1? Despite its somewhat philosophical appearance, the question is a mathematical one. Given some amount of money, in how many ways can you partition it using the available denominations (bank notes and coins)? Today's denominations in CHF are 1000, 200, 100, 50, 20, 10 (banknotes), 5, 2, 1, 0.50, 0.20, 0.10, 0.05 (coins). The amount of CHF 0.20, for example, can be owned in four ways (to get integers, let's switch to centimes): $(20), (10, 10), (10, 5, 5), (5, 5, 5, 5)$. The amount of CHF 0.04 can be owned in no way, while there is exactly one way to own CHF 0.00 (you cannot have 4 centimes in your wallet, but you *can* have no money at all in your wallet).

Solve the problem for any given input amount, by writing a program partition.cpp that defines the following function (all values to be understood as centimes).

```
// PRE: [first, last) is a valid nonempty range that describes
//       a sequence of denominations d_1 > d_2 > ... > d_n > 0
// POST: return value is the number of ways to partition amount
//        using denominations from d_1, ..., d_n
unsigned int partitions (unsigned int amount,
                         const unsigned int* first,
                         const unsigned int* last);
```

Use your program to determine in how many ways you can own CHF 1, and CHF 10. Can your program compute the number of ways for CHF 50? For CHF 100?

Skript-Aufgabe 119 (5 Punkte)

The following function finds an element with a given value x in a sorted sequence (if there is such an element).

```
// PRE: [first, last) is a valid range, and the elements *p,
//      p in [first, last) are in ascending order
// POST: return value is a pointer p in [first, last) such
//       that *p = x, or the pointer last, if no such pointer
//       exists
const int* binary_search (const int* first, const int* last, const int x)
{
  const int n = last - first;
  if (n == 0) return last;          // empty range
  if (n == 1) {
    if (*first == x)
      return first;
    else
      return last;
  }
  // n >= 2
  const int* middle = first + n/2;
  if (*middle > x) {
    // x can't be in [middle, last)
    const int* p = binary_search (first, middle, x);
    if (p == middle)
      return last; // x not found
    else
      return p;
  } else
    // *middle <= x; we may skip [first, middle)
    return binary_search (middle, last, x);
}
```

What is the maximum number $T(n)$ of comparisons between sequence elements and x that this function performs if the number of sequence elements is n? Try to find an upper bound on $T(n)$ that is as good as possible. (You may use the statement of Exercise 120.)

Die **Aufgaben 123** und **124** aus den Vorlesungsunterlagen sind die **Challenge Aufgaben** und geben jeweils 8 Punkte, wenn sie vollständig gelöst werden.

Abgabe: Bis 1. Dezember 2009, 15.15 Uhr.