RWTH Aachen **SS 2012**
Lehrgebiet Theoretische Informatik Problem Set 1
Langer—Reidl—Rossmanith—Sikdar 13.4.2012

Analysis of Algorithms — Tutorial

Problem 1-1

Let x_n, $n \geq 1$ be a solution to the recurrence relation $x_n = \sum_{k=1}^{n} x_k/k$. All solutions form a subvector space of $\mathbf{R}^{\mathbf{N}}$, the space of real sequences. What is the dimension of this subvector space and how does a general solution look like?

Problem 1-2

Consider the following algorithm that computes the maximum element in an array of non-negative integers. We assume that all elements are pairwise different and that each permutation occurs with equal probability.

```
int maxElem(int a[ ], int N) {
  int i, max;

  max = −1;
  for (i = 0; i < N; i++)
    if (a[i] > max)
      max = a[i];
  return max;
}
```

a) How often are the instructions $a[i] > max$ and $max = a[i]$ executed in the *worst case* and in the *best case*?

b) What is the probability that this worst case occurs?

c) How often are the two instructions executed in the *average case*?

Problem 1-3

Let w be a random word in $\{a, b\}^n$ chosen independently and with uniform probability. What is the expected number of iterations of the **while**-loop in the following algorithm? The function *is_palindrome* checks if the given word is a palindrome, i.e., if the word and its reverse are identical.

```
i = 2;
while (i ≤ n)
   if (is_palindrome(w[1], ..., w[i]))
       return true;
   i++;
return false;
```

Homework Assignment 1-1 (10 Points)

Let a be an array of length N, whose entries are random numbers chosen from $\{1, \ldots, N\}$ independently and with uniform probability (i.e., repetitions are possible and likely). What is the expected number of executions of each line of the following algorithm?

```
count = 0;
i = 1;
while (i ≤ N)
    if (a[i]%2 ≡ 1)
        count++;
    i++;
return count;
```

Homework Assignment 1-2 (10 Points)

Two natural numbers $m \neq n$ are called *amicable*, if the sum of all proper factors of m equals n — and the other way around. A son and a father wrote the following programs that compute amicable numbers. What is the running time of the son's program? Find an exact formula for the number of executions of the instruction **if**(a % $i \equiv 0$) as a function of N. Assume that the constant 150000 is replaced by N.

Son

```
#include <iostream>

int e[150000];
int realdiv(int a) {
  int n = 0;
  for(int i = 1; i + i ≤ a; i++)
    if(a%i ≡ 0) n += i;
  e[a] = n;
  return n;
}

main() {
  for(int i = 0; i < 150000; i++) {
    int a = realdiv(i);
    if(a ≥ i) continue;
    if(e[a] ≡ i) std :: cout << i
      << " " << a << "\n";
  }
}
```

Father

```
#include <stdio.h>
#define N 1000000
int factorsum[N];
int main() {
  int i;
  for(i = 1; i < N; i++) {
    int p = i;
    while(p < N) {
      factorsum[p] += i;
      p += i;
    }
  }
  for(i = 1; i < N; i++) {
    int a = factorsum[i] − i;
    if(a < i && i ≡ factorsum[a] − a)
      printf("%d %d\n", a, i);
  }
  return 0;
}
```