RWTH Aachen **SS 2012**
Lehrgebiet Theoretische Informatik Problem Set 3
Langer—Reidl—Rossmanith—Sikdar 30.4.2012

Analysis of Algorithms — Tutorial

Problem 3-1

We analysed Quicksort on inputs that consisted of pairwise distinct keys in random order. Let us now see what happens if the input is ordered in reverse direction.

a) Analyse C_N for the case that the array contains $a[i] = -i$ for $i = 1, \ldots, N$. As always, we suppose that $a[0]$ contains a sentinel element that is smaller than all other keys.

b) Accordingly, analyse E_N.

Problem 3-2

In this exercise, we consider Prim's Algorithm, which computes a minimum spanning tree. The input to this algorithm is a graph $G = (V, E)$, a weight function on the edges $w : E \to \mathbf{R}$ and a starting node r.

```
 1  for each u ∈ V do
 2      key[u] ← ∞
 3      π[u] ← NIL
 4  key[r] ← 0
 5  M ← V
 6  while (M ≠ ∅) do
 7      u ← min-from(M)
 8      for each v ∈ neighbors(u) do
 9          if (v ∈ M) ∧ (w(u, v) < key[v]) then
10              π[v] ← u
11              key[v] ← w(u, v)
```

Construct the control flow graph, a spanning tree in the control flow graph, the fundamental cycles, a corresponding linear system of equations and a solution to this system.

Homework Assignment 3-1 (10 Points)

Quicksort has been analysed by us for different inputs: In the lecture we considered random permutations and in Problem set 3 this week inputs that already ordered — a case that is particularly bad for Quicksort. In this exercise we look at another interesting special case:

a) Analyse C_N for the case that the array contains $a[i] = k$ for $i = 1, \ldots, N$, where k is some natural number (i.e., all keys are identical). We suppose that $a[0]$ contains a sentinel element that is smaller than all other keys, e.g., $a[0] = 0$ or $a[0] = -\infty$.

b) Analyse D_N and E_N under the same conditions. For simplicity, you may assume $N \in \{\, 2^j - 1 \mid j \in \mathbf{N} \,\}$.

Homework Assignment 3-2 (10 Points)

Consider the following program:

```
int selsort(int a[ ], int n) {
    for(int i = 0; i < n; i++) {
        int min = i;
        for(int j = i; j < n; j++) {
            if(a[j] < a[min]) {
                min = j;
            }
        }
        int temp = a[i];
        a[i] = a[min];
        a[min] = temp;
    }
}
```

The input to this program is an array $a[0, \ldots, n-1]$ that contains n pairwise distinct integer keys in random order.

a) Explain how this program sorts the given array.

b) Analyse how often each instruction of the program is executed on average depending on n.

c) There is only one instruction whose analysis is not trivial. Which one is it?

Make a table for small values of n by hand that lists the results for this instruction. Compare the table entries with the results from your closed formula that you obtained in b).