Technische Universität Kaiserslautern

Department of Computer Science
Algorithms & Complexity Group

Issue Date: 07.07.2014
Version: 2014-07-02 18:51

# Exercise Sheet 11 for Algorithm Engineering, SS 14

***Hand In:*** *Until Monday, 14.07.2014, 10:00 am, email to* `wild@cs...` *or in lecture.*

## Problem 25

3 + 2 + 1 points

We consider Quicksort where the pivot is taken from a fixed position of the array. Let us denote by $B_N$ the *expected* number of swaps used to sort a random permutation of length $N$.

a) Argue that any partitioning method requires at least $\frac{1}{6}N - \frac{1}{3}$ swaps in expectation for the *first* partitioning step, not counting the swap potentially needed to bring the pivot to its final place.

b) Show that the overall expected number of swaps $B_N$ is thus at least

$$B_N \quad = \quad \tfrac{1}{3}(N+1)(\mathcal{H}_{N+1} - \tfrac{5}{2}) + \tfrac{1}{2}.$$

c) How many swaps does the pseudocode given in lecture do?

## Problem 26

2 + 4 points

Consider Quicksort with the following alternative partitioning method:

```
Quicksort(A, l, r)
if r ≥ l
    p := A[r]                // Choose rightmost element as pivot
    i := l − 1
    for j = l, ..., r − 1
        if A[j] ≤ p
            i := i + 1
            Swap A[i] and A[j]
        end if
    end for
    i := i + 1
    Swap A[i] and A[r]
    Quicksort(A, l, i − 1)
    Quicksort(A, i + 1, r)
end if
```

a) Argue why Quicksort$(A, 1, n)$ correctly sorts an array $A$ of length $n$.

b) What is the expected number of comparisons and swaps executed by Quicksort$(A, 1, n)$ on an array $A$ that contains a random permutation of $\{1, \dots, n\}$?

   Which Quicksort variant would you prefer?