Albert-Ludwigs-Universität, Inst. für Informatik
Prof. Dr. Fabian Kuhn
H. Ghodselahi, Y. Maus

October 19, 2016

Algorithm Theory, Winter Term 2016/17
Problem Set 1

hand in (hard copy or electronically) by 09:55, Thursday October 27, 2016,
tutorial session will be on October 31, 2016

Exercise 1: Complexity (12 points)

Characterize the relationship between $f(n)$ and $g(n)$ in the following examples by using the O, Θ, or Ω notation. Hence, state for **each** of the following three statements whether $f(n) = \Theta(g(n))$, $f(n) = O(g(n))$ or $f(n) = \Omega(g(n))$ holds.
Explain your answers (formally)!

a) $f(n) = n^2 + 1$ $\qquad$ $g(n) = e^{-n^2}$

b) $f(n) = n^\epsilon$ (for a positive constant $\epsilon < \frac{1}{2}$) $\qquad$ $g(n) = \log_2 n$

c) $f(n) = \log_2(n!)$ $\qquad$ $g(n) = n \log_2 n$

d) $f(n) = \lceil \ln n \rceil !$ $\qquad$ $g(n) = n^2$

Exercise 2: Recurrence Relations (6 points)

a) **(3 points)** Guess the solution of the following recurrence relation by repeated substitution.

$$T(n) \leq 3 \cdot T\left(\frac{n}{3}\right) + c \cdot n \log_3 n, \quad T(1) \leq c$$

where $c > 0$ is a constant.

b) **(3 points)** Use induction to show that your guess is correct.

Remark: You can assume that n equals 3^j for some $j \in \mathbb{N}$.

Exercise 3: Find Local Maximum (22 points)

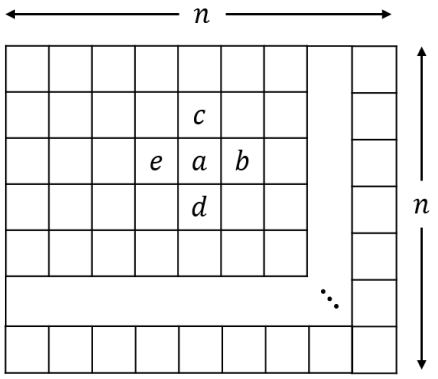

Figure 1: a is a local maximum when a is not smaller than b, c, d, or e.

Consider a one dimensional array $A[1 \dots n]$ such that all values in the array are positive integers. The value $A[i]$ for $i \in \{2, 3, \dots, n-1\}$ is a *local maximum* of the array if

$$A[i-1] \leq A[i] \qquad\qquad A[i+1] \leq A[i] \tag{1}$$

and the value $A[1]$ (resp. $A[n]$) is a local maximum if

$$A[1] \geq A[2] \text{ (resp. } A[n-1] \leq A[n]). \tag{2}$$

In other words, for all $i \in \{1, \dots, n\}$, $A[i]$ is a local maximum if none of its neighboring values are larger.

a) **(4 points)** Prove that any array with positive integers must contain at least one local maximum.

b) **(8 points)** Devise a divide and conquer algorithm to find a local maximum of A in $\mathcal{O}(\log n)$ time.

First you need to formally argue about the correctness of your algorithm (i.e., prove that it computes a local maximum). Second, prove that the runtime of your algorithm is $\mathcal{O}(\log n)$.

c) **(10 points)** Now we want to extend the problem from a one-dimensional array to a two-dimensional array, i.e., an $n \times n$ matrix with positive (integer) entries. Two entries $A[i, j]$ and $A[i', j']$ are neighboring values if $|i - i'| + |j - j'| = 1$.

An entry is a *local maximum* of the matrix if it is not smaller than all its neighboring values (note that an element at the side of the matrix has two or three neighboring values and any other element has four neighboring values). In Figure 1, a is a local maximum if $a \geq b, a \geq c, a \geq d$, and $a \geq e$.

Devise a divide and conquer algorithm that finds a local maximum in a twodimensional $n \times n$ matrix in time $\mathcal{O}(n \log n)$. Again formally prove the correctness of your algorithm and its runtime.