Albert-Ludwigs-Universität, Inst. für Informatik
Prof. Dr. Fabian Kuhn
S. Daum, H. Ghodselahi, O. Saukh October 29, 2013

Algorithm Theory, Winter Term 2013/14
Problem Set 1

hand in by Thursday, November 7, 2013

Exercise 1: Complexity

Characterize the relationship between $f(n)$ and $g(n)$ in the following examples using the O, Θ, or Ω notation. Hence, state if $f(n) = \Theta(g(n))$ and otherwise if $f(n) = O(g(n))$ or $f(n) = \Omega(g(n))$. Explain your answers!

1. $f(n) = n^{0.7}$ $\quad g(n) = 2^{\log_4 n} + n^{0.6} + 5$
2. $f(n) = n^2 + 1$ $\quad g(n) = e^{-n^2}$
3. $f(n) = \ln(n)$ $\quad g(n) = \sum_{i=1}^{n} \frac{1}{i}$
4. $f(n) = \lceil \ln n \rceil !$ $\quad g(n) = n^2$

Exercise 2: Recurrence Relations

In the lecture, we have seen two polynomial multiplication algorithms for which the running times can be described by the recurrence relations $T(n) = 4T(n/2) + c \cdot n$ and $T(1) = c$, as well as $T(n) = 3T(n/2) + c \cdot n$ and $T(1) = c$. Solve the following recurrence relation for a general $q > 2$ and give the specific solutions for $q = 3$ and $q = 4$:

$$T(n) = q \cdot T(n/2) + c \cdot n, \quad T(1) = c.$$

Exercise 3: Almost Closest Pairs of Points

In the lecture, we discussed an $O(n \log n)$-time divide-and-conquer algorithm to determine the closest pair of points. Assume that we are not only interested in the closest pair of points, but in all pairs of points that are at distance at most twice the distance between the closest two points.

1. How many such pairs of points can there be? It is sufficient to give your answer using big-O notation.
2. Devise an algorithm that outputs a list with all pairs of points at distance at most twice the distance between the closest two points. Describe what you have to change compared to the closest pair algorithm of the lecture and analyze the running time of your algorithm.