# Discrete and Computational Geometry, SS 18
# Exercise Sheet "2": Chan's technique/Lower bound
# University of Bonn, Department of Computer Science I

- *Written solutions have to be prepared until* **Tuesday 24th of April**.
- *You may work in groups of at most two participants.*
- *You can hand over your work to our tutor Raoul Nicolodi in the beginning of the lecture.*

**Exercise 4: Constants in Chan's method (4 Points)**

We would like to find out how large the constants in the main lemma of Chan's randomized technique might become. We refer to the proof of Lemma 4 and the application of computing the dilation of a chain.

For the *dilation-of-the-chain* computation we choose $\alpha = \frac{1}{2}$ and $r = 4$ for a decomposition as suggested.

Compute the constant $R$ of Lemma 4 in the following way!

1. Assume that the decision algorithm runs in $C' \times n \log n$ time.
2. Choose an $\epsilon$ so that the precondition of Lemma 4 is fulfilled.
3. How many recursion steps $l$ have to be done for your choice of $\epsilon$?
4. Express constant $C$ in terms of precise values of $l$, $\alpha$ and $r$ and the variable parameter $C'$.

**Exercise 5: Lower bound: Computation of $\hat{y}$ (4 Points)**

This exercise is based on an inequality that was used in the proof for the lower bound on computing the graph-theoretic dilation of a given planar graph $G$. More precisely, the $y$ coordinate of some points had to be chosen such that the inequality holds.

a) Given constants $n, y_\ell, y^u$, provide some value for $y$ depending on $n, y_l, y^u$ that satisfies

$$\left(\frac{y - y_\ell + n + 0.5}{y - y^u + 1}\right)^2 < 1 + \frac{1}{n^2}$$

b) Show that given $a, b > 0$ for each value $\epsilon > 0$ there exists some $\delta > 0$ such that the following holds:

$$1 - \epsilon < \frac{a + \delta}{b + \delta} < 1 + \epsilon$$

**Exercise 6: Randomization arguments (4 Points)**

Let us consider $n$ different natural numbers. Let $q[1], \ldots, q[n]$ denote a random permutation of these numbers. All permutations are chosen with the same probability.

The maximum of $q[1], \ldots, q[n]$ can be computed by the following routine:

$cntr = 0$;
$MaxSoFar = 0$;
for $j = 1$ to $n$ do
  if $(MaxSoFar < q[j])$
    then $MaxSoFar = q[j]$;
      $cntr = cntr + 1$;

How often is $MaxSoFar$ changed in the average? This means you have to give the expected value of the random variable $cntr$ at the end of the routine.

Use the $\Theta$-notation for your result.