Algorithm Theory, Tutorial 1

Johannes Kalmbach

University of Freiburg

johannes.kalmbach@gmail.com

November 2019

General Hints

- Contact tutor (`johannes.kalmbach@gmail.com`) for questions concerning corrections etc.
- Contact forum (`daphne.informatik.uni-freiburg.de`) for everything else
- Suggestion: Submit in groups of two (better for understandable algorithms)
- Submit readable solutions (LaTeX as pdf, CLEAN handwriting (+ good scan if necessary))
- Spend enough time on exercise sheets and writeup (you and I have to understand your submission).

Algorithm Writeups

- Pseudocode, limit to important aspects
- Reader must be able to understand and implement it.
- E.g "Split Array A in two evenly-sized halves L and R"

Shortest Triangle in 2D

- Extend lecture algorithm to triangles
- Complete writeup or only state differences (both is possible)
- Important: what are the pre- and postconditions of recursive algorithms
- Important: Why can we do stuff in a certain runtime

Shortest Triangle in 2D

- Function MinTriangle(P)
  - Return the smallest triangle size
  - Arguments: List of points P
  - Requires / Precondition: P sorted in x-direction
  - Ensures / Postcondition : P sorted in y-direction
- if $|P| \leq$ THRESHOLD:
  - Sort P by y-coordinate (constant)
  - Compute Triangle trivially by n^3 computation on constant input

- if $|P| >$ THRESHOLD:
  - Split P in middle according to x-coordinate (it is sorted!)
  - Recurse on P_l and P_r
  - (We now know the size d of smallest triangle that is completely on one of the two sides)
  - Find all Points closer than d to the border.
  - Build Boxes of size $d \times d$ like in lecture and find triangles in boxes. (Only constant number of points in each box, exactly the same as in the lecture).
  - Merge P_l and P_r according to y-coordinate
  - (P is now sorted by y-coordinate, P_l and P_r were sorted by postcondition of recursive call).
  - Return the smallest triangle.