University of Freiburg
Dept. of Computer Science
Prof. Dr. F. Kuhn
P. Bamberger

Algorithms Theory
Sample Solution Exercise Sheet 1

Exercise 1: Smallest Triangle *(11 Points)*

In the lecture, we discussed an algorithm to determine the distance between the closest pair of points. We now want to solve the following similar problem: Given a set of points S in the plane, determine the size of the smallest triangle. That is, for three pairwise distinct points a, b, c we define $d(a, b, c) := d(a, b) + d(a, c) + d(b, c)$ (where $d(\cdot, \cdot)$ describes the Euclidean distance between two points) and we want to compute $\min\{d(a, b, c) \mid a, b, c \in S \text{ pairwise distinct}\}$.
Describe how to adjust the algorithm from the lecture to solve the given problem. Does the runtime change and if yes, how?

Sample Solution

First sort all points by x-coordinate in $O(n \log n)$ (needs to be done only once, not in each recursion step). The algorithm works as follows:

- If $|S| = 3$, i.e., $S = \{a, b, c\}$ for some distinct points a, b, c, return $d(a, b, c)$ (if $|S| < 3$, return ∞).
- Divide S into two equal sized sets S_ℓ and S_r.
- Recursively compute Δ_ℓ and Δ_r as the size of the smallest triangle in S_ℓ and S_r and recursively sort S_ℓ and S_r according to y-coordinates.
- Combine: Merge to sort S according to y-coordinates (as in the mergesort algorithm). Return $\min\{\Delta_\ell, \Delta_r, \Delta_{\ell r}\}$ with $\Delta_{\ell r} = \min\{d(x, y, z) \mid \text{one point in } S_\ell, \text{ one point in } S_r\}$.

We explain the combine step: Let $\Delta := \min\{\Delta_\ell, \Delta_r\}$. Let x_0 be the median of all x-coordinates. We only need to consider so-called center points with an x-coordinate that is within distance $\leq \Delta/2$ of x_0. We go through these points in order of increasing y-coordinates. For each point s, we need to check the sizes of the triangles that s forms with any two other center points (where at least one is on the other side) which have a y-coordinate that is at most $\Delta/2$ larger than that of s. All these points lie in a rectangle R of size $\Delta \times \Delta/2$. We can partition R into 18 squares of size $\Delta/6$, either lying full on the left or full on the right side. Within such a square, each two points have distance $< \Delta/3$. Therefore, at most two points can lie in the same square (because three points in one square would form a triangle of size $< \Delta$). We have to check the triangles that s builds with any pair of points in $R \cap S_r$ which are $\leq 18^2 - 18$ many and the triangles that s builds with one point in $R \cap S_\ell$ and one point in $R \cap S_r$ which are $\leq 18^2$ many. So overall, we have to check at most $630 = O(1)$ triangles for s. It follows that the combine step takes $O(n)$. The runtime analysis is therefore the same as for the closest pair of points.

Exercise 2: Landau-Notation *(3+3+3 Points)*

Prove or disprove the following statements

(a) $4n^3 + 8n^2 + 5n \in O(2n^3)$.

(b) $2n \in O(10\sqrt{n})$.

(c) $\log_2(2^n \cdot n^3) \in \Theta(5n)$

Sample Solution

(a) True. For all n we have $n^3 \geq n^2 \geq n$ and thus $4n^3 + 8n^2 + 5n \leq 17n^3 \leq 9 \cdot 2n^3$ (i.e. choose $n = 1$ and $c = 9$ in the definition of the O-notation).

(b) False. Let $f(n) = 2n$ and $g(n) = 10\sqrt{n}$. Let $c > 0$. We have $f(n) \leq c \cdot g(n) \Leftrightarrow n \leq 25c^2$. So for any $c > 0$ and any n_0, there is an $n \geq n_0$ with $f(n) > c \cdot g(n)$ (for given c and n_0 choose $n = \max\{n_0, \lceil 25c^2 \rceil + 1\}$).

(c) True. We have $\log_2(2^n \cdot n^3) = \log_2(2^n) + \log_2(n^3) = n + 3\log_2(n)$. As $\log_2(n) \leq n$ for all $n \geq 1$ we have $n + 3\log_2(n) \leq 4n \leq 5n$ (i.e. choose $n = 1$ and $c = 1$ in the definition of the O-notation).