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

Algorithms Theory
Sample Solution Exercise Sheet 1

Due: Tuesday, 10th of November 2020, 4 pm (*please note the information given on page 2*)

Exercise 1: Landau-Notation *(3+3+3 Points)*

Prove or disprove the following statements.

(a) $n^3 - 9n^2 \in \Omega(n^3)$.

(b) $\big(\log(\sqrt{n})\big)^2 \in \Theta(\log n)$.

(c) $n^n \in \Omega\big((2n)!\big)$.

Sample Solution

(a) True. For all $n \geq 18$ we have $n^3 \geq 18n^2$ and thus $2(n^3 - 9n^2) = n^3 + (n^3 - 18n^2) \geq n^3$. i.e., $n^3 - 9n^2 \geq \frac{1}{2}n^3$ (choose $c = 1/2$ and $n_0 = 18$).

(b) False. We show $(\log(\sqrt{n}))^2 \notin O(\log n)$. We have $(\log(\sqrt{n}))^2 = \frac{1}{4}(\log n)^2$ and thus for $c > 0$

$$\begin{array}{rccc} & (\log(\sqrt{n}))^2 & \leq & c\log n \\ \Leftrightarrow & (\log n)^2 & \leq & 4c\log n \\ \Leftrightarrow & \log n & \leq & 4c \\ \Leftrightarrow & n & \leq & 2^{4c} \end{array}$$

For given $c, n_0 > 0$ choose $n := \max\{\lceil 2^{4c}\rceil + 1, n_0\}$. Then we have $n \geq n_0$ but $(\log(\sqrt{n}))^2 > c\log n$.

(c) False. We have

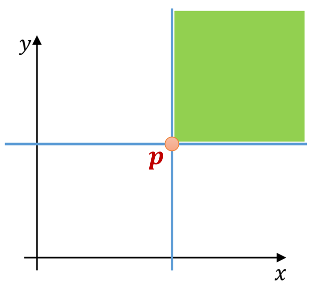
$$(2n)! = 2n \cdot (2n-1) \cdot (n+1) \cdot n! > n^n \cdot n! \geq n^n \cdot n\ .$$

Hence, for given $c, n_0 > 0$, choose $n := \max\{\lceil c\rceil, n_0\}$. Then we have $n \geq n_0$ but

$$c \cdot n^n < (2n)!\ .$$

Exercise 2: Divide-and-Conquer *(11 Points)*

Consider a set of n points in the plane given by their x and y coordinates ($x, y \geq 0$). We say that $p = (x, y)$ is a *Pareto optimal* point if for all other points $p' = (x', y')$ it holds $x' < x$ or $y' < y$. That is, a point p is *Pareto optimal* if in the following picture, the area marked green is empty:



Give a divide-and-conquer algorithm that finds all *Pareto optimal* points in time $O(n \log n)$. Analyze the runtime.

Sample Solution

We present two ways of solving:

I.

Algorithm 1 `pareto`(S) ▷ Input S: set of points

1: **if** $|S| = 1$ **then**
2:     **return** S
3: **else**
4:     Compute the median m of S w.r.t. the lexicographic order.
5:     $S_\ell = \{q \in S \mid q <_{lex} m\}$ ▷ split S into left and right part
6:     $S_r = \{q \in S \mid q \geq_{lex} m\}$
7:     $P_\ell =$ `pareto`(S_ℓ)
8:     $P_r =$ `pareto`(S_r)
9:     $y_0 = \max\{y \mid \exists x : (x, y) \in S_r\}$
10:     $P'_\ell = \{(x, y) \in P_\ell \mid y > y_0\}$
11:     **return** $P'_\ell \cup P_r$

Runtime: Dividing (computing the median) costs $O(n)$ and combining (lines 9-11) $O(n)$ as well. The recurrence relation is hence $T(n) = 2T(n/2) + O(n)$ with base case $T(1) = O(1)$, which yields a runtime of $O(n \log n)$.

II.
Given a set S of points, we use mergesort to sort the points lexicographically which takes time $O(n \log n)$. Having S as a sorted array, we proceed in the following way:

Algorithm 2 `pareto1`(S)

Allocate an empty array P of length n ▷ Array of Pareto optimal points
$i = n - 1$
$y_{max} = -1$
while $i \geq 0$ **do**
    $q = S[i]$
    $y = q[1]$ ▷ q is a point (tuple) with $q[0]$ its x-coordinate and $q[1]$ its y-coordinate
    if $y > y_{max}$ **then**
        $P.append(q)$
        $y_{max} = y$
    $i - -$
return P

To see the correctness of the algorithm, consider a point q in the lexicographically sorted array S. All points to its left have either a smaller x-coordinate or the same x-coordinate and a smaller y-coordinate. This means that q "wins" all comparisons with its points to the left. To win the comparison with a point to its right, it needs to have a larger y-coordinate. Hence, q wins all comparisons if its y-coordinate is larger than the maximum y-coordinate of the points to its right.
The runtime of `pareto1` is $O(n)$, so combined with the time for sorting we obtain an overall runtime of $O(n \log n)$.

Submission Information

Format: This semester we will only accept digital submissions! All submissions must be in, or converted to pdf format. We strongly recommend to prepare your solutions with Latex for best readability. Solutions prepared with Word or similar text editors are ok. Scans or photos of handwritten solutions in pdf format are ok as well, but must be well readable!

Submission Guidelines: The exercises will be conducted online with the course management system Daphne. The solution of each exercise **must** be uploaded to your SVN repository each one in a separate folder named *exercise-XY*, where XY is the exercise number (with a leading 0 if that number is smaller than 10). More on the submission via SVN on our website.

Team Submission: Teams will be allowed. Teams may have at most 3 members! In case you submit your solution as part of a team, each team member must still submit a copy of the solution pdf to their respective SVN-repository (c.f. explanations on our website). The members of the teams must be clearly marked on the top of the solution pdf.