Technische Universität München
Fakultät für Informatik
Lehrstuhl für Algorithmen und Komplexität
Prof. Dr. Harald Räcke
Richard Stotz

Winter term 2018/19
Assignment 2
October 22, 2017

Efficient Algorithms and Data Structures I

Deadline: October 29, 10:15 am in the ***Efficient Algorithms*** *mailbox.*

Homework 1 (5 Points)

The biologist Andrew wants to determine, whether the unicorn he found is real or fake. Fake unicorns can be detected based on their number of colors N in their mane. Andrew knows that the following algorithm checks if a unicorn is fake:

Algorithm 1: UnicornCheck(N)

```
for i = 2 ... N − 1 do
    if N mod i == 0 then
        return Unicorn is fake!
return Unicorn is real!
```

(a) Show that the worst-case running time of the algorithm in the uniform cost model is $\mathcal{O}(N)$.

(b) Assume that computing $p \bmod q$ takes time $\lfloor (p/q) \log p \rfloor$ in the logarithmic cost model. Show that the worst-case running time of the algorithm in the logarithmic cost model is $\mathcal{O}(N(\log N)^2)$.

(c) Argue for both models that the running time of algorithm UnicornCheck(N) is not polynomial in the input size.

Homework 2 (6 Points)

1. Show that $n^{\ln n} \in o((\ln n)^n)$.

2. Show that $n^{\ln \ln \ln n} \in o(\lceil \ln(n) \rceil !)$.

3. Show that $F^2_{\lceil H_n \rceil} \in o\left(H_{F_n}\right)$, where $H_n = \sum_{i=1}^{n} \frac{1}{i}$ and F_n is the nth Fibonacci number.

Hints: Use $\ln n \leq H_n \leq \ln n + 1$ and the closed-form representation of the Fibonacci numbers.

Homework 3 (4 Points)

Let $f, g : \mathbb{N} \to \mathbb{R}^+$ be two positive monotonically increasing functions.
Prove or disprove the following statements. Use precise arguments based on the definition of the Landau-notation shown in the lecture.

1. For any positive, monotone increasing function $f : \mathbb{R} \to \mathbb{R}$, it holds that $f(\log_2(n)) \in \Theta(f(\log_4(n)))$.

2. $f(n) \in \Theta(f(n/4))$.

Homework 4 (5 Points)

Let log denote the binary logarithm. The function $\log^{(i)} n$ is defined inductively by

$$\log^{(i)} n = \begin{cases} n & \text{if } i = 0 \\ \log(\log^{(i-1)} n) & \text{if } i > 0 \end{cases} .$$

The *iterated logarithm function* $\log^* n$ describes the number of logarithms that need to be applied in order to reduce n to 1. Formally,

$$\log^* n = \min\{i \geq 0 \mid \log^{(i)} n \leq 1\} .$$

1. Compute $\log^* 4$ and $\log^*(2^{65536})$.

2. Describe a function tower : $\mathbb{N} \to \mathbb{R}$ such that $\log^*(\text{tower}(n)) = n$.

3. Which is asymptotically larger: $\log(\log^* n)$ or $\log^*(\log n)$?

Bonus Homework 1 (6 Bonus Points)

Bonus homework can be used to improve the overall score for the bonus.
We also award small prizes for well-written solutions.
During an excavation in a temple near Alexandria, archeologist Anton uncovers a scroll with a curious algorithm. Sadly, the code is not documented, as comments were only invented a few centuries later.

Algorithm 2: Strange(n)

$X \leftarrow \{(i, n-i) \mid i = 1, \ldots, n-1\}$
while $\max_{(a,b)\in X} b > 0$ **do**
  $X \leftarrow \{(|a-b|, \min\{a,b\}) \mid (a,b) \in X\}$
return $\max_{(a,b)\in X} a$

(a) What does the algorithm compute? Prove your claim!

(b) Prove that the algorithm runs in $\mathcal{O}(n^2)$.

(c) Prove that the algorithm does *not* run in $o(n \log^k n)$ for any constant k.

Tutorial Exercise 1

Late in autumn, the squirrel Alexander wants to sort all nuts that he collected over the summer by their size. He uses the traditional algorithm SQUIRREL-SORT (Algorithm 3).

Algorithm 3: SQUIRREL-SORT(A, i, j)

```
if (A[i] > A[j]) then
    swap A[i] ↔ A[j]
if i + 1 ≥ j then
    return
k ← ⌊(j − i + 1)/3⌋
SQUIRREL-SORT(A, i, j − k)
SQUIRREL-SORT(A, i + k, j)
SQUIRREL-SORT(A, i, j − k)
```

1. Argue that *SQUIRREL-SORT*$(A, 1, n)$ correctly sorts a given array $A[1 \dots n]$. Use induction over the array length.

2. Analyze how much time Alexander asymptotically needs to sort his n nuts using a recurrence relation.

```
I feel as if I should succeed in doing something in
mathematics,
although I cannot see why it is so very important ...
                                          - H. Keller
```