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

Winter term 2019/20
Assignment 02
October 21, 2019

Efficient Algorithms and Data Structures I

*Deadline: October 28, 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 $i = 2 \dots N-1$ **do**
    if $N \bmod i == 0$ **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 $\Omega(N)$.

(b) Suppose 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 $\Omega(N(\log N)^2)$.

(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 $\ln n \le H_n \le \ln n + 1$ and the closed-form representation of the Fibonacci numbers.

Homework 3 (4 Points)

Show by using the basic definition of the Θ-notation, that for any real constants α and β, where $\beta > 0$,

$$(n-\alpha)^{\beta} = \Theta(n^{\beta}) \ .$$

Homework 4 (5 Points)

Suppose $f, g : \mathbb{N} \to \mathbb{R}^{+}$ are two positive functions with $f(n) \in \mathcal{O}(g(n))$. Prove or disprove the following statements:

(a) $f(n) \in \mathcal{O}(g(n)/10)$,

(b) $\ln(f(n)) \in \mathcal{O}(\ln(g(n)))$ if $f(n) > 1$ and $g(n) > 1$ for all $n > 0$,

(c) $2^{f(n)} \in \mathcal{O}(2^{g(n)})$.

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 2).

Algorithm 2: SQUIRREL-SORT(A, i, j)

if $(A[i] > A[j])$ **then**
    swap $A[i] \leftrightarrow A[j]$
if $i + 1 \geq j$ **then**
    return
$k \leftarrow \lfloor (j - i + 1)/3 \rfloor$
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.

```
friend, canst thou analyse this in thy mind,
and of these masses all the measures find,
go forth in glory!  be assured all deem
thy wisdom in this discipline supreme!
                                  - Archimedes
```