Institute of
Pervasive Computing

Algorithms and Data Structures 1
Summer term 2023
Jäger, Beck, Zafar

Deadline CS: **Tue. 04.04.2023, 08:00**
Deadline AI: **Thu. 06.04.2023, 08:00**
Submission via: **Moodle**

Assignment 1

Elaboration time

Remember the time you need for the elaboration of this assignment and do the survey in Moodle.

COMPLEXITY / RECURRENCES

For this assignment, please submit one high-quality PDF file that contains the solution to all examples.

1. Finding Complexities 1+2+3 points

Determine the run-time complexity for the worst-case scenario of the following algorithms in big O notation – also give arguments for your solution approach!
Note: The algorithms are imaginary and possibly not optimized. "# Statements" parts are constant.

a)
```
# n > 0

for i in range(n):
    # Statements
    for j in range(i, n):
        # Statements
# Statements
```

b) `# m, n > 0, n < m <` n^2
```
i = 1
while i < n:
    j = 1
    while j < n:
        # Statements
        j += 1
    k = m
    while k > m:
        # Statements
        k -= 1
    i *= 10
i = 1
while i < n:
    # Statements
    i += 10
```

c)
```
# a, b, c > 0

if a < b and b < c:
    for i in range(a):
        # Statements
    if c < a:
        for j in range(c):
            # Statements
    else:
        for k in range(b):
            # Statements
elif a > b and b > c:
    for i in range(c, b):
        # Statements
else:
    for i in range(a, a + 5):
        # Statements
```

2. Unfolding + Proofing 3+3 points

a) Solve the following recurrence using **unfolding**:

$$T(1) = 1$$
$$T(n) = 100T(n/10) + n$$

b) Proof (using **guess and proofing**) that your result from 2a is correct.

3. Master Theorem 4+4+4 points

Solve the following recurrence using the **Master Theorem**:

a) $T(n) = 8T(n/2) + n^3$
b) $T(n) = T(n/2) + n * log\, n$
c) $T(n) = 3T(n/3) + log\, n$

Always provide your approach and verify the additional condition for case 3 ($\exists \varepsilon > 0 \colon f(n) = \Omega(n^{\log_b a + \varepsilon})$).