36	37	38	39	40

Problems Solved:

Name:

Matrikel-Nr.:

Problem 36. Define *concrete* languages L_i ($i = 1, \dots, 4$) over the alphabet $\Sigma = \{0, 1\}$ such that L_i has infinitely many words and $L_i \neq \Sigma^*$. The following properties must be fulfilled.

(i) There exists (deterministic) Turing machine M_1 with $L_1 = L(M_1)$ such that every word $w \in L_1$ is accepted in $O(1)$ steps.

(ii) Every (deterministic) Turing machine M_2 with $L_2 = L(M_2)$ needs at least $O(n)$ steps to accept a word $w \in L_2$ with $|w| = n \in \mathbb{N}$.

(iii) Every (deterministic) Turing machine M_3 with $L_3 = L(M_3)$ needs at least $O(n^2)$ steps to accept a word $w \in L_3$ with $|w| = n \in \mathbb{N}$.

(iv) Every (deterministic) Turing machine M_4 with $L_4 = L(M_4)$ needs at least $O(2^n)$ steps to accept a word $w \in L_4$ with $|w| = n \in \mathbb{N}$.

By *concrete* language it is meant that your definition defines an explicit set of words (preferably of the form $L_i = \{w \in \Sigma^* \mid \dots\}$) and not simply a class from which to choose. In other words,

> Let $L_1 \neq \Sigma^*$ be an infinite language such that (i) holds.

does not count as a *concrete* language.
In each case (informally) argue why your language fulfills the respective conditions.
Note that the exercise asks about acceptance of a word, not the computation of a result.

Problem 37. Determine the asymptotic time and space complexity of the program depending on the input N (use Θ-notation).

```
n = read()
p = 1
while n > 0
    p = 2 * p
    n = n - 1
q = 1
while p > 0
    q = 2 * q
    p = p - 1
write(q)
```

Note: The time complexity is considered to be the number of lines executed, and the space complexity is the number of variables used during the execution.

Problem 38. Is there a Turing machine M over the alphabet $\Sigma = \{0, 1\}$ that can multiply two 64 bit integers in $O(1)$ steps? The Turing machine would get the binary representations of two integers, i. e., two words of length 64, as input and has to produce the product in binary form as output. Justify your answer.

If the answer is *yes*, describe how the multiplication is done and find an upper bound for the number of steps, if the answer is *no*, explain why it is not possible.

Problem 39. True or false?

1. $(2n+3)(3n+2) = O(n^2)$
2. $(2n+3) + \log_2(3n^6+2) = O(n)$
3. $\frac{1024}{2^n} = O(1)$
4. $\frac{1024}{2^n} = \Theta(\log_2(n))$
5. $4^n = O(2^n)$
6. $2^n = O(4^n)$

Prove your answers based on Definition 45 from the lecture notes.

Problem 40. Prove or disprove the following:

1. $O(g(n))^2 = O(g(n)^2)$
2. $2^{O(g(n))} = O(2^{g(n)})$

Hint: First transform above equations into a form that does not involve the O-notation on the left hand side, then prove the correctness of the resulting formulas.