Problems Solved:

36	37	38	39	40

Name:

Matrikel-Nr.:

Problem 36.

1. Consider the probability space $\Omega = \{0,1\}^n$ of all strings over $\{0,1\}$ of length n where each string occurs with the same probability 2^{-n}. Let $X : \Omega \to \mathbb{N}$ be a random variable that gives the position of the first occurrence of the symbol 1 in a string, if 1 occurs at all. For completeness, we also define that $X(0^n) = 0$. Positions are numbered from 1 to n. Give a (not necessarily closed form, i. e., the solution may use the summation sign) term for the expected value $E(X)$ of the random variable X and justify your answer.

2. Evaluate the sum
$$S = \sum_{k=1}^{n} \frac{1}{2^k} k$$
in *closed form*, i.e., find a formula for the sum which does not involve a summation sign.

   Hint: Take the function
$$F(z) := \sum_{k=0}^{n} \left(\frac{z}{2}\right)^k.$$
and let $F'(z)$ denote the first derivative of $F(z)$. We then have $S = F'(1)$. Why?

   Thus, it suffices to compute a closed form of $F(z)$, using your high-school knowledge about geometric series. Then compute the first derivative $F'(z)$ of this form, and, finally, evaluate $F'(1)$.

   Note that the index for the geometric series starts at $k = 0$.

Problem 37. Let $M = (Q, \Gamma, \sqcup, \Sigma, \delta, q_0, F)$ be a Turing machine with $Q = \{q_0, q_1\}$, $\Sigma = \{0,1\}$, $\Gamma = \{0,1,\sqcup\}$, $F = \{q_1\}$ and the following transition function δ:

δ	0	1	$\sqcup$
q_0	$q_0 0 R$	$q_1 1 R$	–
q_1	–	–	–

1. Determine the (worst-case) time complexity $T(n)$ and the (worst-case) space complexity $S(n)$ of M.

2. Determine the average-case time complexity $\bar{T}(n)$ and the average-case space complexity $\bar{S}(n)$ of M. (Assume that all 2^n input words of length n occur with the same probability, i.e., $1/2^n$.)

3. Bonus: Using results from Problem 36, express all answers in closed form, i.e., without the use of the summation symbol.

Problem 38. Write a LOOP program in the core syntax (variables may be only incremented/decremented by 1) that computes the function $f : \mathbb{N} \to \mathbb{N}$, $f(n) = 2^n$.

1. Count the number of variable assignments (depending on n) during the execution of your LOOP program with input n.
2. What is the time complexity of your program (depending on n)?
3. Is it possible to write a LOOP program with time complexity better than $O(2^n)$? Give an informal reasoning of your answer.
4. Let $l(k)$ denote the bit length of a number $k \in \mathbb{N}$. Let $b = l(n)$, i.e., b denotes the bit length of the input. What is the time complexity of your program depending on b, if every variable assignment $x_i := x_j + 1$ costs time $O(l(x_j))$?

Problem 39. True or false?

1. $5n^2 + 7 = O(n^2)$
2. $5n^2 = O(n^3)$
3. $4n + n \log n = O(n)$
4. $(n \log n + 1024 \log n)^2 = O(n^2 (\log n)^3)$
5. $3^n = O(9^n)$
6. $9^n = O(3^n)$

Prove your answers based on the formal definition of $O(f(n))$, i. e., for all functions $f, g : \mathbb{N} \to \mathbb{R}_{\geq 0}$ we have

$$g(n) = O(f(n)) \iff \exists c \in \mathbb{R}_{>0} : \exists N \in \mathbb{N} : \forall n \geq N : g(n) \leq c \cdot f(n).$$

Problem 40. Show by formal proof based on the definition of O-notation that for all functions $f, g, h : \mathbb{N} \to \mathbb{R}_{\geq 0}$ the following holds:
If $f \in O(g)$ and $g \in O(h)$, then $f \in O(h)$.