Problems Solved:

41	42	43	44	45

Name:

Matrikel-Nr.:

Problem 41.

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 term (not necessarily in closed form, i. e., the solution may use the summation sign) 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 $F'(1)$.

   You may *check* your result with the help of a computer algebra system or `https://www.wolframalpha.com`. Note, however, that simply writing down what the computer algebra system gives you is only counted, if it comes along with a proof that the function that you called gives exactly what is asked for in this problem together with a proof that this function is implemented without bugs.

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

Solution of Problem 41:

1. $X(0^{k-1}1x_{k+1}\dots x_n) = k$ and this case occurs with probability $2^{n-k}2^{-n} = 2^{-k}$. So the expected value is
$$E(X) = \sum_{k=1}^{n} k \cdot 2^{-k} + 0 \cdot 2^{-n} = \sum_{k=1}^{n} \frac{k}{2^k}$$

2. Let

$$F(z) = \sum_{k=0}^{n} \left(\frac{z}{2}\right)^k.$$

Then

$$F'(z) = \sum_{k=1}^{n} \frac{k \cdot z^{k-1}}{2^k} = \sum_{k=1}^{n} \frac{k}{2^k} z^{k-1}.$$

So $S = F'(1)$. By the geometric series,

$$F(z) = \frac{\left(\frac{z}{2}\right)^{n+1} - 1}{\frac{z}{2} - 1} = \frac{2\left(2^{-n-1}z^{n+1} - 1\right)}{2\left(\frac{z}{2} - 1\right)} = \frac{2^{-n}z^{n+1} - 2}{z - 2}.$$

By the quotient rule $\left(\frac{u}{v}\right)' = \frac{u'}{v} - \frac{uv'}{v^2}$, we get

$$F'(z) = \frac{(n+1)2^{-n}z^n}{z-2} - \frac{2^{-n}z^{n+1} - 2}{(z-2)^2}$$
$$F'(1) = -(n+1)2^{-n} - (2^{-n} - 2) = 2 - \frac{n+2}{2^n}$$

Problem 42. 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 41, express all answers in closed form, i.e., without the use of the summation symbol.

Solution of Problem 42:

1. The worst case occurs for the input word 0^n. The machine M makes exactly one move to the right for each 0 read. Therefore,

$$T(n) = S(n) = n.$$

2. Since M moves always to the right, $\bar{S}(n) = \bar{T}(n)$. The machine M stops after having found the first occurrence of 1 in the input. Therefore, the number of moves it makes depends on the position k of the first

occurrence of 1 in the input (if it occurs at all): An input word of the form $0^{k-1}1(0+1)^{n-k}$ causes M to make exactly k moves and such a word occurs with probability $1/2^k$. (We assume that all 2^n input words occur with the same probability $1/2^n$). The input word 0^n causes M to make n moves and it occurs with probability $1/2^n$. Therefore,

$$\bar{S}(n) = \bar{T}(n) = \sum_{k=1}^{n} \frac{1}{2^k}k + \frac{1}{2^n}n.$$

3. By using $\sum_{k=1}^{n} \frac{1}{2^k}k = 2 - \frac{n+2}{2^n}$ from Problem 41

$$\bar{S}(n) = \bar{T}(n) = 2 - \frac{n+2}{2^n} + \frac{n}{2^n} = 2 - \frac{2}{2^n}.$$

Note that $\bar{T}(n) < 2$ for all n.

Problem 43. Let M be a Turing machine over the alphabet $\{0,1\}$ that takes as input a string $b_1b_2\dots b_n$ ($b_i \in \{0,1\}$), prepends an additional 1 to the string and then interprets the result $1b_1b_2\dots b_n$ as the binary representation of a number k. M then writes out the unary representation of k (consisting of a string of k letters 1) onto the tape and stops.
Note that in the above description it is not said *how* M computes the result. In particular M need not be the most efficient Turing machine fulfilling the above specification.

1. Give a reasonably close asymptotic lower-bound for the space- and time-complexity $S(n)$ and $T(n)$ for the execution of the task and justify these bounds (without giving a detailed construction of M). Choose adequate Landau-symbols for formulating the bounds.
2. Give an informal description of a (reasonably efficient) Turing machine M' that performs the task described above. Analyze the space and time complexity $S(n)$ and $T(n)$ and write down an upper/exact asymptotic bound for these complexities. Again choose adequate Landau symbols for formulating the bounds.

   Hint: Let M' apply the *binary powering* strategy.

Solution of Problem 43:

1. (a) $\Omega(2^n)$. We must write between 2^n and $2^{n+1}-1$ letters 1 onto the tape.
   (b) $\Omega(2^n)$. We must write between 2^n and $2^{n+1}-1$ letters 1 onto the tape. The time complexity is never better than the space complexity. Of course, it is hard to imagine that $\Omega(2^n)$ is the biggest lower bound (see the second part of the task). Anyway $\Omega(2^n)$ would count as a sufficient solution.
2. (a) The TM can use "binary powering" to complete its job.
   (i) First mark the first tape cell, to be able to find the beginning of the tape. I.e. turn 0 into $\bar{0}$ and 1 into $\bar{1}$.

(ii) Go to the first blank and write a 1 after it.
(iii) Go back to the marked cell.
(iv) If the marked cell was originally a blank, then move all the 1's after the first blank to the beginning of the tape and stop.
(v) Otherwise, remember the contents of the marked cell in the state.
(vi) If the marked cell contained 0, then double the 1's after the first blank. If the cell contained 1, then double the 1's after the first blank and add an additional 1 after this result.
(vii) Continue with the next unmarked cell.

Space complexity is clearly $\Theta(2^n)$.

(b) Time complexity is a bit harder. Essentially there is a loop in the above description of the TM. Assuming for simplicity that the initial string was 0^n then this leads (in the i-th iteration to the expression
  i. $2(n-i)$ to go over the bits that are left from the input
  ii. $2^i \cdot 2^i \cdot c$ to double the 1's after the blank, for the copy of each of the 2^i 1's, one has to to $c \cdot 2^1$ steps (where c is some constant).

Thus we are led to compute $\sum_{i=1}^{n}(2(n-i)+c\cdot 4^i)$. Ignoring the "linear" part and using geometric series, we find $O(4^n)$ as a time complexity for an "efficient" TM.

Problem 44. Let X be a monoid. Device an "algorithm" (as recursive/iterative pseudo-code in the style of Chapter 6 of the lecture notes) for the computation of x^n for $x \in X, n \in \mathbb{N}$ that uses less multiplications than the naive algorithm of n times multiplying x to the result obtained so far. Determine the complexity as $M(n)$, i.e., the number of multiplications of your "algorithm" depending on the exponent n.
Hint: Note that x^8 can be computed with just 3 multiplications while the naive algorithm would use 7 multiplications. Based on this observation, the algorithm can be based on a kind of "binary powering" strategy.

Solution of Problem 44:

$$x^n = \begin{cases} 1 & \text{if } n = 0 \\ (x^{n/2})^2 & \text{for even } n > 0 \\ x(x^{(n-1)/2})^2 & \text{for odd } n > 0 \end{cases}$$

Let $M(n)$ be the number of multiplications performed by the above algorithm for the computation of x^n then

$$M(n) = \begin{cases} 0, & \text{if } n = 0 \text{ or } n = 1, \\ M(n/2) + 1, & \text{for even } n > 0 \\ M((n-1)/2) + 2, & \text{for odd } n > 1 \end{cases}$$

Thus $M(n) \le M(n/2) + 2 \le M(n/4) + 4 \le \dots \le M(n/2^k) + 2k$.

For $k = \min_k(n < 2^k)$ we have $M(n) \le M(0) + 2k = 2k$ and $2^{k-1} \le n \implies k \le \log_2(n) + 1 \implies k = O(\log n)$.

Thus $M(n) = 2O(\log n) = O(\log n)$.

Problem 45. Let $T(n)$ be given by the recurrence relation

$$T(n) = 3T(\lfloor n/2 \rfloor).$$

and the initial value $T(1) = 1$. Show that $T(n) = O(n^\alpha)$ with $\alpha = \log_2(3)$.
Hint: Define $P(n) :\iff T(n) \le n^\alpha$. Show that $P(n)$ holds for all $n \ge 1$ by induction on n. It is not necessary to restrict your attention to powers of two.

Solution of Problem 45:

Define $P(n) :\iff T(n) \le n^\alpha$. We prove $P(n)$ for all $n \ge 1$ by induction on n.

Induction base: $P(1)$ holds: $1 \le 1$.

Induction step: We can prove the bound

$$T(n) = 3T(\lfloor n/2 \rfloor) \le 3\lfloor n/2 \rfloor^a \le 3(n/2)^a = \frac{3}{2^a} n^a = n^a.$$

which gives $T(n) \le n^a$. The first equality is from the recurrence. The following $\le$ comes from the induction hypothesis, which says that $P(k)$ holds for all $k < n$, and so in particular $P(\lfloor n/2 \rfloor)$ holds. The last equality is from $2^{\log_2 3} = 3$.

By the induction principle, $T(n) \le n^\alpha$ holds for all $n \ge 1$.