Problems Solved:

41	42	43	44	45

Name:

Matrikel-Nr.:

Problem 41. Analyze the (worst-case) time and space complexity of a Turing maschine which computes the sum of two numbers. The input $(k, m) \in \mathbb{N} \times \mathbb{N}$ is encoded as $1^k 0 1^m$ and trailed by $\sqcup$'s.
Note that you are expected to provide an explicit definition of the TM that is analyzed.

Problem 42. Let $T(n)$ be the total number of times that the instruction `a[i,j] = a[i,j] + 1` is executed during the execution of `P(n,0,0)`.

```
procedure P(n, x, y)
    if n >= 1 then
        for (i = x; i < x+n; i++)
            for (j = y; j < y+n; j++)
                a[i,j] = a[i,j] + 1
        h = floor( n / 2)
        P(h, x,   y  )
        P(h, x+h, y  )
        P(h, x,   y+h)
        P(h, x+h, y+h)
    end if
end procedure
```

1. Compute $T(1)$, $T(2)$ and $T(4)$.
2. Give a recurrence relation for $T(n)$.
3. Solve your recurrence relation for $T(n)$ in the special case where $n = 2^m$ is a power of two.
4. Use the Master Theorem to determine asymptotic bounds for $T(n)$.

Problem 43. Given two algorithms A and B for computing the same problem. For their time complexity we have

$$t_A(n) = \sqrt{n} \quad \text{and} \quad t_B(n) = 2^{\sqrt{\log_2 n}}.$$

1. Construct a table for $t_A(n)$ and $t_B(n)$. Can you give a value N such that for all $n \geq N$ one of the algorithms always seems faster than the other one?
2. Based on your result of the question above, you may conjecture $t_A(n) = O(t_B(n))$ and/or $t_B(n) = O(t_A(n))$. Prove your conjecture(s) formally on the basis of the O notation.

Hint: remember that for all $x, y > 0$ we have

$$x = 2^{\log_2 x}$$

$$\log_2 x^y = y \cdot \log_2 x$$
$$\sqrt{x} = x^{\frac{1}{2}}$$
$$x \le y \;\Rightarrow\; 2^x \le 2^y$$

which may become handy in your proof.

Problem 44.

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 $F'(1)$.

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

Problem 45. 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 44, express all answers in closed form, i.e., without the use of the summation symbol.