Problems Solved:

41	42	43	44	45

Name:

Matrikel-Nr.:

Problem 41. Let $\Sigma = \{0, 1\}$ and let $L \subseteq \Sigma^*$ be the set of binary numbers divisible by 3, i.e.,

$$L = \{x_n \dots x_1 x_0 \ : \ 3 \text{ divides } \sum_{k=0}^{n} x_k 2^k\}.$$

(By convention, the empty string ε denotes the number 0 and so it is in L too.)

1. Design a Turing machine M with input alphabet Σ which recognizes L, halts on every input, and has (worst-case) time complexity $T(n) = n$. Write down your machine formally. (A picture is not needed.) *Hint:* Three states q_0, q_1, q_2 suffice. The machine is in state q_r if the bits read so far yield a binary number which leaves a remainder of r upon division by 3. The transition from one state to another represents a multiplication by 2 and the addition of 0 or 1.
2. Determine $S(n)$, $\bar{T}(n)$ and $\bar{S}(n)$ for your Turing machine.
3. Is there some faster Turing machine that achieves $\bar{T}(n) < n$? (Justify your answer.)

Problem 42. Let $T(n)$ be the number of multiplications carried out by the following Java program.

```
int a, b, i, product, max;
max = 1;
a = 0;
while ( a < n ) {
  b = a;
  while (b <= n) {
    product = 1;
    i = a;
    while (i < b) {
      product = product * factors[i];
      i = i + 1; }
    if (product > max) { max = product; }
    b = b + 1; }
  a = a + 1; }
```

1. Determine $T(n)$ exactly as a nested sum.
2. Determine $T(n)$ asymptotically using Θ-Notation. In the derivation, you may use the asymptotic equation

$$\sum_{k=0}^{n} k^m = \Theta(n^{m+1}) \text{ for } n \to \infty$$

for fixed $m \geq 0$ which follows from approximating the sum by an integral:

$$\sum_{k=0}^{n} k^m \simeq \int_0^n x^m \, dx = \frac{1}{m+1} n^{m+1} = \Theta(n^{m+1})$$

Problem 43. Let $T(n)$ be total number of calls to `tick()` resulting from running `P(n)`.

```
procedure P(n)
    k = 0
    while k < n do
        tick()
        P(k)
        k = k + 1
    end while
end procedure
```

1. Compute $T(0), T(1), T(2), T(3), T(4)$.
2. Give a recurrence relation for $T(n)$. (It is OK if your recurrence involves a sum.)
3. Give a recurrence relation for $T(n)$ that does not involve a sum. (*Hint:* Use your recurrence relation (twice) in $T(n+1) - T(n)$.)
4. Solve your recurrence relation. (It is OK to just guess the solution as long as you prove that it satisfies the recurrence.)

Problem 44. 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 $P(n) :\Longleftrightarrow T(n) \leq n^\alpha$. Show that $P(n)$ holds for all $n \geq 1$ by induction on n. It is not necessary to restrict your attention to powers of two.

Problem 45. 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 $T(n)$.