Problems Solved:

41	42	43	44	45

Name:

Matrikel-Nr.:

Problem 41. 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 for n after which 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

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

which may become handy in your proof.

Problem 42. 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^k01^m and trailed by $\sqcup$'s.
Note that you are expected to provide an explicit definition of the TM that is analyzed.

Problem 43. 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 44. 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 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 $P(n) :\iff T(n) \leq n^\alpha$. 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.