Problems Solved:

41	42	43	44	45

Name:

Matrikel-Nr.:

Problem 41. 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 42. 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 43. 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.

Problem 44. 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 45. Consider the following Java methods:

```
int a(int n) { return b(n); }
int b(int n) {
  if (n<1) {return -1;}
  int p = b(n-1);
  int q = b(n-2);
  int r = b(n-2);
  return p+q+r; }
```

Let f_n be the number of calls to method `b` which result from evaluating `a(`n`)` where $n \geq 0$. (We assume that no optimizations are made. In particular, *both* of the function calls `b(n-2)` in lines 5 and 6 are executed.)

1. Compute f_n for $n = 0, 1, \ldots, 5$.

2. Give a recurrence relation for f_n.

3. Let
$$F(z) = \sum_{n=0}^{\infty} f_n z^n = 1 + z + 4z^2 + \ldots$$
be the *generating function* of the sequence f_n. (Assume that this sum converges; for $|z| < 1/2$ it does.) Show that the generating function $F(z)$ satisfies the equation
$$F(z) = zF(z) + 2z^2F(z) + \frac{1}{1-z} - z. \tag{1}$$
Hint: Multiply both sides of your recurrence equation by z^n; then sum it on n. Rewrite your resulting sums in terms of $F(z)$.

4. Solve the equation above for $F(z)$ an perform a partial fraction decomposition on your result. *Hint:* The correct result is
$$F(z) = \frac{1}{1-2z} + \frac{1}{2} \cdot \frac{1}{1+z} - \frac{1}{2} \cdot \frac{1}{1-z}.$$
Your answer should show your calculation to get this result.

5. Find a closed form expression (i.e. a formula) for f_n.
Hint: Apply the geometric series
$$\frac{1}{1-az} = \sum_{k=0}^{\infty} a^k z^k.$$
to each summand in the solution for $F(z)$ in Part 4 with $a = 2$, $a = 1$ and $a = -1$ and bring the result in a form that matches
$$F(z) = \sum_{n=0}^{\infty} f_n z^n$$
in order to find f_n.