Problems Solved:

46	47	48	49	50

Name:

Matrikel-Nr.:

Problem 46. Let $L = \left\{ ww^{-1} \,\middle|\, w \in \{0,1\}^* \right\}$ be the language of palindromes.

- Describe (informally) a Turing machine M with $L(M) = L$.
- Analyse the time and space complexity of M.

Problem 47. Consider the following pseudo code of an implementation of a FIFO (first in first out) queue.

```
input  := EMPTYLIST
output := EMPTYLIST
function enqueue(e, input, output) { push(e, input) }
function dequeue(input, output) {
    if isempty(output) {
        while not isempty(input) { push(pop(input), output) }
    }
    pop(output)
}
```

Analyze its amortized cost by (a) the aggregate method and (b) the potential method.
Here,

- `push(e, L)` is the operation of adding an element e to the front of a (singly linked) list L,
- `isempty(L)` returns `TRUE` if the list L is empty,
- `pop(L)` is the operation that removes the first element of a list L and returns it.

All these operations are assumed to cost constant time.
In the code above, a queue is represented by the pair (`input`, `output`). Putting a new element into the queue via `enqueue`, first puts it to the front of an intermediate list (called `input`). Only when an element is requested via a call to `dequeue`, elements are moved from the `input` to the `output` list, thus effectively reversing the `input` list so that in total the queue returns its elements in a FIFO principle.
Hint: For the potential method you might want to consider the function Φ such that for a queue q that is represented by the pair (`input`, `output`) of two lists, $\Phi(q)$ is the size of the `input` list.

Problem 48. Consider the program

```
f(n) ==
    return g(n, 0, 0, 0)
g(n, m, v, s) ==
    if m > n then
        return s
    else
        return g(n, m + 1, 2 * (v + (2 * m + 1) * 2^m), s + v)
```

which computes a function $f : \mathbb{N} \to \mathbb{N}$.

1. Show that
$$v = 2^m m^2$$
holds true for every call $g(n, m, v, s)$ to function g in the execution of $f(n)$. *Hint:* Use induction on the number of (nested) function calls to g.

2. Show that
$$s = \sum_{k=0}^{m-1} k^2 2^k$$
holds true for every call $g(n, m, v, s)$ in the execution of $f(n)$.

   Hint: Again, use induction on the number of calls to g. In the induction step, you may want to use the result of Part 1.

3. From Part 2, one may deduce $f(n) = \sum_{k=0}^{m} k^2 2^k$.

   Show by induction on n that $f(n) = 2^{n+1}(3 - 2n + n^2) - 6$.

Problem 49. Let M be a Turing machine that stops on every input and let L be a finite language over the alphabet $\{0, 1\}$. Does there exist a Turing machine D that decides whether $L \subseteq L(M)$ holds. Justify your answer.

Problem 50. Consider a RAM program that evaluates the value of $\sum_{i=1}^{n} i^2$ in the naive way (by iteration). Analyze the worst-case asymptotic time and space complexity of this algorithm on a RAM assuming the existence of operations `ADD r` and `MUL r` for the addition and multiplication of the accumulator with the content of register r.

1. Determine a Θ-expression for the number $S(n)$ of registers used in the program with input n (space complexity).

2. Compute a Θ-expression for the number $T(n)$ of instructions executed for input n (time complexity in constant cost model),

3. Assume a simplified version of the logarithmic cost model of a RAM where the cost of every operaton is proportional to the length of the arguments involved. In particular, if a is the (bit) length of the accumulator and l is the (bit) length of the content of register r then `MUL r` costs $a + l$ and `ADD r` costs $\max(a, l)$.

   Compute the assymptotic costs $C(n)$ (using O-notation) of the algorithm for input n.