Seminar Complexity & Algorithms – week 1

During seminar a selection of the next exercises will be practiced and discussed.
The rest of the exercises are recommended to use for extra practice material.

Exercise 1
Compute the next logarithms:

a) ${}^{2}\log 128$
b) ${}^{2}\log 2048$
c) ${}^{5}\log 125$
d) ${}^{3}\log 81$
e) ${}^{10}\log 1000000$
f) ${}^{7}\log 49$
g) $\log 16$
h) $\log 256$
i) $\log 1$
j) $\log (1024^2)$
k) $\log (16*32)$
l) $\log (\log 65536)$

Exercise 2
Give an approximation of the next logarithms:

a) log 10
b) log 20
c) log 30
d) log 100
e) log 200
f) log 300
g) log 1000
h) log 2000
i) log 3000

Exercise 3
We want to download a file from the internet. Initially there is a delay of 5 seconds (for making the connection), and then the downloading starts at 2,5 MB/sec. Give a formula for the time needed to download a file of N MB via this connection.

Exercise 4
Recently a judge convicted a municipality because of negligence to pay a fine of 2 euros on the first day. On every next day the negligence still consists, the fine will be doubled (so the fine becomes in succession 2, 4, 8, 16, 32, 64, … euros).

a) What is the fine at day N?
b) After how many days the fine will be at least D euros?

Exercise 5
Compute a big-oh expression for the running time T(n), for each of the next program fragments:

```
1) sum = 0;
   for ( i = 0; i < n; i++ )
       sum++;

2) sum = 0;
   for ( i = 0; i < n; i++ )
       for ( j = 0; j < n; j++ )
           sum++;

3) sum = 0;
   for ( i = 0; i < n; i++ )
       for ( j = 0; j < n*n; j++ )
           sum++;
```

```
4) sum = 0;
   for ( i = 0; i < n; i++ )
       for ( j = 0; j < i; j++ )
           sum++;

5) sum = 0;
   for ( i = 0; i < n; i++ )
       for ( j = 0; j < i*i; j++ )
           for ( k = 0; k < j; k++ )
               sum++;

6) sum = 0;
   for ( i = 1; i < n; i++ )
       for ( j = 1; j < i*i; j++ )
           if ( j % i == 0 )
               for ( k = 0; k < j; k++ )
                   sum++;
```

Exercise 6
Compute a big-oh expression for the running time T(n), for each of the next program fragments:

```
1) sum = 0;
   for (i = 0; i < n; i += 2)
       sum++;

2) sum = 0;
   for (i = 0; i < n; i++)
       sum++;
   for (j = 0; j < n; j++)
       sum++;

3) sum = 0;
   for (i = 1; i < n; i = i * 2)
       sum++;
```

Exercise 7
An algorithm needs 0,5 ms to process an input of 100 elements. How long will it take approximately to process 1000 elements, when the running time of the algorithm

a) is linear?
b) is O(N log N)?
c) is quadratic?
d) is cubic?

Exercise 8
An algorithm hneeds 0,5 ms to process an input of 100 elementen. How many elements can approximately be processed by the algorithm in 2 minutes, when the running time

a) is linear?
b) is quadratic?
c) is cubic?

Exercise 9

a) What is the running time of insertionsort when the input is already sorted? And when the input is sorted in reverse order?
b) What is the running time of quicksort when the input is already sorted and when the pivot is the first element? And when thee pivot is the middle element?
c) What is the running time of mergesort when the input is already sorted? And when it is sorted in reverse order?

Exercise 10

Given is a sequence of numbers. We are looking for the smallest difference between any two numbers that are in this sequence.
Example: sequence = { 1, 15, 27, 12, 33, 56, 7} → smallest difference = 3 (i.e. 15-12)
Solution 1: find the smallest difference by computing the differences of each pair of numbers.
Solution 2: sort the sequence and find the smallest difference by computing the differences of each pair of successive numbers.

a) Give (pseudo-)code for both solutions.
b) Give an estimation of the big-Oh for each of these two solution methods.

Exercise 11 – Sorting in linear time

When you have more information on the items to be sorted, then it is possible to sort them in linear time. Show that a sequence of N 16-bit integers can be sorted in O(N) time.
Hint: Use an array, indexed from 0 till 65535.