Theoretical Computer Science – Exercise 11

SS 2022
Jochen Schmidt

Technische
Hochschule
Rosenheim

Please prepare the following exercises at home prior to the tutorial:

Exercise 1

To solve a given problem, an algorithm A and an algorithm B are available. The execution times are $t_A = 2n^2$ and $t_B = 8n^{1.75}$, where n is are the number of input data. It is reasonable to switch between

Exercise 2

A sorting algorithm needs exactly 1ms to sort 1000 data records. The time T(n) required by some sorting algorithm to sort n data records is directly proportional to n log n, i.e., T(n) = c n log n. The time to sort N data records has been measured and is given by T_N.

a) Specify a formula for T(n) as a function of T_N. Does the base of the logarithm matter?

b) Let T_N = 1ms to sort 1000 data records. Calculate the time it takes to sort 1,000,000 records.

Exercise 3

a) An algorithm with complexity $O(n^2)$ requires 2ms to process 400 data records. How long does it take for 8000 records?

b) An algorithm needs 10s to process 1000 data records. How long does it take for 100,000 records if the complexity is O(n)? How long for complexity $O(n^3)$?

Exercise 4

Each of the expressions below specifies the computation time for an algorithm to solve a size n problem. Enter the complexity order in O-notation.

Expression	O(…)
$5 + 0{,}001n^3 + 0{,}025n$	
$100n + 20n^{1{,}5} + 10n \log_{10} n$	
$n^2 \log_2 n + n(\log_2 n)^2$	
$100000n + 10n^2$	
$n \log_3 n + n \log_2 n$	
$0{,}03 \log_4 n + \log_2 \log_2 n$	

Exercise 5

Given are two algorithms A and B, which require $T_A(n) = 5n \log_{10} n$ and $T_B(n) = 25n$ microseconds for a problem of size n.

a) Which algorithm is the better one in terms of O-notation?

b) From what amount of data does the better performance apply?

Exercise 6

The obvious algorithm for calculating x^n requires n – 1 multiplications. Specify a faster **recursive** algorithm for the special case that the exponent is a power of two, i.e., $n = 2^m$, and calculate its complexity in O-notation (the number of multiplications required is counted here).

We will do the following exercises together during the tutorial:

Exercise 7

Determine the complexity of the following code snippets in O-notation (n is the size of the data):

```
a) for( int i = n; i > 0; i /= 2 )
   {
     for( int j = 1; j < n; j *= 2 )
     {
       for( int k = 0; k < n; k += 2 )
       {

          ... // constant number of operations
        }
      }
    }

b) for( int bound = 1; bound <= n; bound *= 2 )
   {
     for( int i = 0; i < bound; i++ )
     {
       for( int j = 0; j < n; j += 2 )
       {
         ... // constant number of operations
       }
       for( int j = 1; j < n; j *= 2 )
       {
         ... // constant number of operations
       }
     }
   }
```