Data structures and algorithms
Exercises for course lecture notes

English version by Viktória Vadon
based on [1, 2]

Contents

1 Mathematics introduction **1**

2 Number representations **1**

3 Recursion **3**

4 Order symbols and the "master" theorem **3**

1 Mathematics introduction

Exercise 1.1 (Floor, ceil, round and fractional part)**.** Complete the table:

x	-4	-3.8	-3.5	-3.2	-3	3	3.2	3.5	3.8	4
$\lfloor x \rfloor$										
$\lceil x \rceil$										
$\mathrm{Round}(x)$										
$\{x\}$										

Exercise 1.2 (Floor, ceil, round and fractional part functions)**.** Plot the floor, ceil, round and fractional part functions!

Exercise 1.3 (div and mod operations)**.** Calculate ± 12 div ± 5 and ± 12 mod ± 5 for all possible combination of signs!

2 Number representations

Exercise 2.1 (Number in different bases)**.** Each row should contain the same number in different bases.

a) Complete the table!

base						
2	3	4	8	10	12	16
10110110						
	120102					
		30201				
			7421			
				2024		
					8A4B	
						5F03

b) Complete the table!

base						
2	3	4	8	10	12	16
101110.011						
	12012.12					
		302.01				
			742.15			
				2024.25		
					8A4B.36	
						5F03.84

Exercise 2.2 (Number of digits)**.** Using the theorem on the number of digits, without making the conversion, predict how many digits a number requires in a different base!

a) Complete the table!

		number of digits		
		base		
		2	10	16
number	2^2			
	10^{10}			
	16^{16}			

b) A number is 34 digits in base two. How many digits is it in base sixteen? Can we generalize a rule?

c) A number is 9 digits in base sixteen. How many digits is it in base two? Can we generalize a rule?

Exercise 2.3 (Floating point numbers)**.** a) Represent the following (base ten) numbers as single precision floating point numbers! What are the resulting hexadecimal bytes?

$$1387.1875 \qquad -1208.6875 \qquad 625.90625$$

b) The following hexadecimal bytes represent single precision floating point numbers. What are the numbers (in base ten)?

```
C4 69 26 00      44 5B 0E 00      C4 BE 9A 00
```

3 Recursion

Exercise 3.1 (Factorial 1)**.** For $n \in \mathbb{Z}^+$, we define n factorial, $n! := 1 \cdot 2 \cdot \ldots \cdot n$, the product of positive integers up to n. Alternatively, we can use a recursive definition

$$n! = \begin{cases} (n-1)! \cdot n, & \text{if } n > 1, \\ 1, & \text{if } n = 1. \end{cases}$$

Suppose we have a procedure that calculates the factorial based on the recursive formula, and it was initially called to calculate 7!. How does the stack evolve during the run and calculation? What is the largest depth? How many *recursive* calls are made?

Exercise 3.2 (Factorial 2)**.** More generally, let us define the product of consecutive numbers from $m \leq n \in \mathbb{Z}^+$ as

$$P(m,n) := \begin{cases} m, & \text{if } m = n, \\ P\left(m, \left\lfloor \frac{m+n}{2} \right\rfloor\right) \cdot P\left(\left\lfloor \frac{m+n}{2} \right\rfloor + 1, n\right), & \text{if } m < n. \end{cases}$$

Suppose we have a procedure that can calculate $P(m,n)$ based on this recursion, and it is called to calculate $7! = P(1,7)$. How does the stack evolve during the run and calculation? What is the largest depth? How many *recursive* calls are made?

Exercise 3.3 (Binomial coefficient)**.** For $k \leq n \in \mathbb{N}$, the binomial coefficient "n choose k" is defined recursively as

$$\binom{n}{k} := \begin{cases} 1, & \text{if } k = 0, \\ 1, & \text{if } k = n, \\ \binom{n-1}{k-1} + \binom{n-1}{k}, & \text{if } 0 < k < n. \end{cases}$$

Suppose we have a procedure that calculates *binomnk* based on this recursion, and it is called to calculate $\binom{7}{3}$. How does the stack evolve during the run and calculation? What is the largest depth? How many *recursive* calls are made?

Exercise 3.4 (Fibonacci)**.** The Fibonacci numbers, for indices $n \in \mathbb{N}$, are defined recursively as

$$F(n) := \begin{cases} 0, & \text{if } n = 0, \\ 1, & \text{if } n = 1, \\ F(n-2) + F(n-1), & \text{if } n > 1. \end{cases}$$

Suppose we have a procedure that calculates $F(n)$ based on this recursion, and it is called to calculate $F(7)$. How does the stack evolve during the run and calculation? What is the largest depth? How many *recursive* calls are made?

4 Order symbols and the "master" theorem

Exercise 4.1 (Relations between order symbols)**.** Prove the following statements:

a) If $f(n) = o\big(g(n)\big)$, then also $f(n) = O\big(g(n)\big)$.

b) Is part a) reversible? Why, or why not?

c) If $f(n) = \omega\big(g(n)\big)$, then also $f(n) = \Omega\big(g(n)\big)$.

d) Is part c) reversible? Why or why not?

e) If $f(n) = O\big(g(n)\big)$ and $f(n) = \Omega\big(g(n)\big)$, then $f(n) = \Theta\big(g(n)\big)$.

f) Is part e) reversible? Why or why not?

Exercise 4.2 (Order symbols)**.** Are the following statements true?

a) $2n^2 + 3n = O(n^2)$

b) $2n^2 + 3n = o(n^2)$

c) $2n^2 + 3n = O(n^3)$

d) $2n^2 + 3n = O(n)$

e) $2n^2 + 3n = \Omega(n^2)$

f) $2n^2 + 3n = \omega(n^2)$

g) $2n^2 + 3n = \Omega(n^3)$

h) $2n^2 + 3n = \Omega(n)$

i) $2n^2 + 3n = \Theta(n^2)$

Exercise 4.3 (Order symbols 2)**.** Are the following statements true?

a) $\log(n) = O(n)$

b) $\log(n) = o(n)$

c) $\log(n) = O(n^\varepsilon)$ for any or all $\varepsilon \in \mathbb{R}^+$

d) $\log(n) = o(n^\varepsilon)$ for any or all $\varepsilon \in \mathbb{R}^+$

e) $\log(n) = \Omega(n^\varepsilon)$ for any or all $\varepsilon \in \mathbb{R}^+$

f) $\log(n) = \omega(n^\varepsilon)$ for any or all $\varepsilon \in \mathbb{R}^+$

g) Is $\log(n)$ polynomially faster growing than $1 = n^0$?

h) Is $\log(n)$ polynomially slower growing than n?

Exercise 4.4 ("Master" theorem)**.** Consider the following, recursively defined functions. Can we determine their growth order using the master theorem? If yes, what is their growth order?

a) $T(n) = 2 \cdot T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + \log(n)$

b) $T(n) = 2 \cdot T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + 2n + 3$

c) $T(n) = 2 \cdot T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + n \cdot \log(n)$

d) $T(n) = 2 \cdot T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + n^{3/2}$

References

[1] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2022.

[2] Attila Házy and Ferenc Nagy. *Adatstruktúrák és algoritmusok*. https://www.uni-miskolc.hu/~matha/adat_alg_NF_HA.pdf.