Exercise 2.1  Recurrence relations.

Find a closed form for recurrence relations of the form

\[ T(n) = \begin{cases} 
  aT\left( \frac{n}{b} \right) + cn + d & \text{falls } n > 1 \\
  e & \text{falls } n = 1 
\end{cases} \]

with \( a, b, c, d, e \in \mathbb{N}, b > 1 \). Prove your answer using mathematical induction. You can assume that \( n \) is a power of \( b \).

Note: In your proof, consider the cases i) \( a \neq b, a \neq 1 \) ii) \( a \neq b, a = 1 \) and iii) \( a = b \).

Exercise 2.2  Estimating asymptotic running time.

Specify (as concisely as possible) the asymptotic running time of the following code fragment in \( \Theta \) notation depending on \( n \in \mathbb{N} \). You do not need to justify your answer.

---

a) 1 \textbf{for}\( (\text{int } i = 0; i < n; i++) \)
2 \textbf{for}\( (\text{int } j = 0; j < i/2; j++) \)
3 ;

b) 1 \textbf{for}\( (\text{int } i = 1; i <= n*n; i++) \)\
2 \textbf{int } k = i*i;
3 \textbf{while}(k > 1)
4 \hspace{1em} k = k/4;
5 }

---

c) 1 \textbf{int } f(\text{int } n) \{ 
2 \hspace{1em} \textbf{if}(n == 1) \textbf{return} 1;
3 \hspace{1em} \textbf{else} \{ 
4 \hspace{2em} \textbf{for}(\text{int } i = 1; i <= n; i++) 
5 \hspace{2em} ;
6 \hspace{1em} \textbf{return} f(n/3)+1;
7 \}
8 \}
Exercise 2.3 Various topics.

a) The integer multiplication method of Karatsuba/Ofman computes the product of two numbers recursively using a formula that, except for additions and multiplications with the base (here: 10), contains three products. Give two numbers $x$ and $y$ such that these products are $(15 \cdot 86)$, $(87 \cdot 72)$ and $(15 \pm 72) \cdot (87 \pm 86)$.

$$x = \text{________________} \quad y = \text{________________}$$

b) Give a sequence of 5 numbers such that bubble sort needs exactly 10 swaps to sort it.

Answer: _____, _____, _____, _____, _____

c) Provide the key pairs that are considered in the first four comparisons when the following array is sorted with insertion sort.

\[
\begin{array}{cccccccccc}
8 & 15 & 7 & 12 & 4 & 9 & 10 & 11 & 13 & 19 \\
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10
\end{array}
\]

Key pairs: (____, ____), (____, ____), (____, ____), (____, ____)

Exercise 2.4 Lower bounds / Algorithm design.

Consider a set of $n$ coins that contains exactly one false one. This false coin is heavier than all other ones. To find the false coin you can only use a balance scale. Using this you can only determine whether the coins on the left are lighter, heavier or have exactly the same weight as the coins on the right.

a) Construct a strategy for $n = 9$ coins that uses as few weighings as possible.

b) For a general $n$, provide an algorithm that uses in the worst case exactly $\log_3(n)$ many weighings.

c) Show that even the best algorithm will use at least $\log_3(n) - 1$ many weighings in the worst case.

*Note:* You can assume that $n$ is a power of 3.