# Tutorial 2 Exercises

COMP526: Efficient Algorithms

14–15 October, 2024

**Exercise 1.** Consider the sequence of numbers $T(n)$ defined recursively by

$$T(n) = \begin{cases} 3 & \text{if } n = 0; \\ T(n-1) + 4 & \text{if } n \geq 1. \end{cases}$$

(a) Compute the first 6 elements of $T(n)$, i.e., $T(0)$, $T(1)$, $T(2)$, $T(3)$, $T(4)$, and $T(5)$.

(b) Make an educated guess about the general pattern that this sequence follows. Write this guess as a *closed form* for $T(n)$, i.e., a formula for $T(n)$ without recursive reference to $T$.

(c) Now formally prove the correctness of your guess using mathematical induction.

**Exercise 2.** Recall that given positive integers $n$ and $k$, the ***modulo operation*** $n \bmod k$ computes the remainder when $n$ is divided by $k$. That is, $r = n \bmod k$ if and only if $n = q \cdot k + r$ for some integer $q$ and $0 \leq r < k$. Consider the following MOD procedure that computes $n \bmod k$.

```
procedure Mod(n, k)
    t ← n
    while t ≥ k do
        t ← t − k
    end while
    return t
end procedure
```

(a) Argue that MOD$(n,k)$ correctly computes $n \bmod k$. (Hint: what is a loop invariant maintained after each iteration of the loop?)

(b) Express the running time of this procedure as a function of $n$ and $k$ using big-O notation.