Weekly Assignment 1: Big-O

September 2022

Some facts (no exercises)

Some tips and tricks (most of these are not needed for this particular exercise sheet, but might come in handy at some point during the course). Summations:

$$1 + 2 + \cdots + n = \sum_{i=0}^{n} i = \frac{n(n+1)}{2} \qquad 1 + r + \cdots + r^n = \sum_{i=0}^{n} r^i = \frac{r^{n+1} - 1}{r - 1}$$

If these are not familiar to you, try to prove them yourself (for example with induction), practicing such proofs is always useful. About logarithms and exponentials:

$$2^0 = 1 \qquad 2^1 = 2 \qquad 2^a 2^b = 2^{a+b} \qquad 2^{ab} = (2^a)^b = (2^b)^a$$

$$\log_2(1) = 0 \qquad \log_2(2) = 1 \qquad \log_2(ab) = \log_2(a) + \log_2(b)$$

If the logarithm rules are unfamiliar to you, try proving them from the exponentiation rules and the identity:

$$\log_2(2^x) = x \qquad \text{for all } x \in \mathbb{R}$$

1. Suppose we have a computer which can perform 1 million ($= 10^6$) operations per second. The seven formulas below denote the running time of some algorithms (measured in number of operations) depending on the number of elements n we feed to the algorithm. Determine for each algorithm how many elements can be processed in 1 minute.
   (a) n^2
   (b) $n \log n$
   (c) 2^n
   (d) $n\sqrt{n}$
   (e) n^{100}
   (f) 4^n
   (g) n
   (h) n^3
   (i) $n!$
2. Determine whether the following statements are true or false, no proof is required.
   (a) $2n \in \mathcal{O}(n)$
   (b) $14n^2 + 4n + 18 \in \mathcal{O}(n^2)$

(c) $n + \log(n) \in \mathcal{O}(n)$

(d) $2^{2n} + 230 \in \mathcal{O}(2^n)$

(e) $n! \in \mathcal{O}(2^n)$

(f) $n! \in \mathcal{O}(n^n)$

(g) $n\sqrt{n} \in \mathcal{O}(n \log(n))$

3. Recall that to prove $f \in \mathcal{O}(g)$ one has to find a $c > 0$ and $n_0 \in \mathbb{N}$ such that $f(n) \leq c \cdot g(n)$ for all $n \geq n_0$.

   Prove, using the definition above, that:

   (a) $n + 1 \in \mathcal{O}(n)$

   (b) $n^3 + n + 2 \in \mathcal{O}(n^3)$

   (c) If $f \in \mathcal{O}(g)$ and $g \in \mathcal{O}(h)$, then $f \in \mathcal{O}(h)$

   (d) It is not the case (i.e. give a counterexample) that if $f \in \mathcal{O}(h)$ and $g \in \mathcal{O}(h)$, then $f \cdot g \in \mathcal{O}(h)$

4. The factorial of a non-negative integer n, denoted by $n!$, is defined as follows

$$\begin{cases} 0! = 1 \\ n! = n \cdot (n-1)! \end{cases}$$

   Write a recursive function returning the factorial of n and determine its asymptotic time complexity.

5. Provide the asymptotic time complexity of the following algorithms, justify your answers. You may assume that every operation takes a single time step, except **end while** which takes none.

(a)

Algorithm 1

```
Require: n
s ← 0
i ← 0
while i < n do
    s ← s + 1
    i ← i + 1
end while
```

(b)

Algorithm 2

```
Require: n
s ← 0
i ← 0
while i < n do
    j ← 0
    while j < n do
        s ← s + 1
        j ← j + 1
    end while
    i ← i + 1
end while
```

Algorithm 3

(c) **Require:** n

```
s ← 0
i ← 0
while i < n do
    j ← 0
    while j < n · n do
        s ← s + 1
        j ← j + 1
    end while
    i ← i + 1
end while
```

Algorithm 4

(d) **Require:** n

```
s ← 0
i ← 0
while i < n do
    j ← 0
    while j < i do
        s ← s + 1
    end while
    i ← i + 1
end while
```

Algorithm 5

(e) **Require:** n

```
s ← 0
i ← 0
while i < n do
    j ← 0
    while j < i do
        k ← 0
        while k < j do
            s ← s + 1
            k ← k + 1
        end while
        j ← j + 1
    end while
    i ← i + 1
end while
```