

Exercise/Problem Guide for *Nature of Computation* by Moore and Mertens

Muthu Chidambaram

Last Updated: June 20, 2019

Contents

1	Appendix	3
1.1	The Story of O	3
1.1.1	Exercise A.1	3
1.1.2	Exercise A.2	3
1.1.3	Exercise A.3	3
1.1.4	Exercise A.4	3
1.1.5	Exercise A.5	3
1.1.6	Exercise A.6	3
1.1.7	Exercise A.7	3
1.1.8	Exercise A.8	3
1.1.9	Exercise A.9	3

About

“Computer science is no more about computers than astronomy is about telescopes.” - Edsger Dijkstra (Maybe)

These notes contain my solutions to some exercises and problems from the book *Nature of Computation* by Christopher Moore and Stephan Mertens.

1 Appendix

1.1 The Story of O

1.1.1 Exercise A.1

We can choose $n_3 = \max(n_1, n_2)$ such that $f_1(n) + f_2(n) \leq (C_1 + C_2)g$ whenever $n > n_3$.

1.1.2 Exercise A.2

Similar to the first exercise, but now we have $C_1 C_2 h$ instead.

1.1.3 Exercise A.3

Logarithms of different bases differ by a constant.

1.1.4 Exercise A.4

The argument treats k as a constant, when in reality $k = O(n)$. The answer should be $O(n^3)$, which can be checked by looking at the closed form of the sum.

1.1.5 Exercise A.5

$2^{O(n)}$ and $O(2^n)$ are not the same, since $\limsup_{n \rightarrow \infty} \frac{2^{C_1 n}}{C_2 2^n}$ only converges to a nonzero constant when $C_1 = 1$.

1.1.6 Exercise A.6

$n^k \neq \Theta(2^n)$, $e^{-n} \neq \Theta(n^{-c})$, and $n! \neq \Theta(n^n)$, as all limits go to 0.

1.1.7 Exercise A.7

Take $f = n$ and $g = 2n$.

1.1.8 Exercise A.8

1. We can pull out the exponents as constants, so $f = \Theta(g)$.
2. $3 < 2^2$ so $f = o(g)$.
3. $n = \omega(\log^2 n)$ so $f = \omega(g)$.
4. $2^{\log n} \rightarrow \infty$ so $f = \omega(g)$.

1.1.9 Exercise A.9

One example is $n^{\log n}$.