Solutions to
Algorithms by Dasgupta, Papadimitriou, Vazirani

Arkaprava Choudhury

January 3, 2021

Preliminary note

This is just a rough sketch of the reasoning I used to solve the exercises found in the book Algorithms by Dasgupta, Papadimitriou, Vazirani. At the time of writing this, I have only solved the exercises in Chapter 0, so I do not know whether I will be able to complete all the exercises.

I should also address the fact that I am self-studying this book in preparation of later courses. I figured that since I will need to learn this material at some point in time, it would be better to gain a head-start and familiarize myself with these terms. It is winter break of my first year currently. This project is simply a means to avoid letting my time go waste.

Huge kudos to the work done by Raymond Feng from UC Berkeley whose solutions I referred occasionally whenever I got stuck with a question for far too long. Check out his GitHub for the rest of the solutions, if this document is not complete by the time you are reading this.

1. Chapter 0 Exercises

Exercise 0.1.

0.1.a. $f \in \Theta(g)$

0.1.b. $f \in \mathcal{O}(g)$

0.1.c. Firstly, note that the Big-O notation can also be expressed in the form of limits to infinity. For example, if we have two functions $u, v : \mathbb{R}^{\geq 0} \to \mathbb{N}$ such that $u \in \Theta(v)$, and some nonzero positive constant $c \in \mathbb{R}^{+}$, then we conclude:

$$\lim_{x\to\infty} \frac{u(x)}{v(x)} = c \qquad \text{(using definitions)}$$

Thus, applying this definition, we now see:

$$\begin{aligned}
\lim_{n\to\infty} \frac{n + (\log n)^2}{n} &= \lim_{n\to\infty} \frac{n}{n} + \lim_{n\to\infty} \frac{(\log n)^2}{n} \\
&= 1 + \lim_{n\to\infty} \frac{2\log n}{n} && \text{(using L'Hopital's Rule)} \\
&= 1 + 0 && \text{(using L'Hopital's Rule again)} \\
&= 1 \\
\lim_{n\to\infty} \frac{100n + \log n}{n} &= \lim_{n\to\infty} \frac{100n}{n} + \lim_{n\to\infty} \frac{\log n}{n} \\
&= 100 + 0 && \text{(using L'Hopital's Rule again)} \\
&= 100
\end{aligned}$$

Thus, $f, g \in \Theta(n)$, and so, we also have, $f \in \Theta(g)$.

0.1.d. $g = 10n \cdot (\log n + \log 10)$. Thus, $f \in \Theta(g)$.

0.1.e. $f \in \Theta(g)$.

0.1.f. $g = 2\log n$. Thus, $f \in \Theta(g)$.

0.1.g. Firstly, note that for all $n \geq 10.273$, we have, $\log^2 n \geq n^{0.01}$. Thus, we have, for all $n \geq 10.273$, $n\log^2 n \geq n^{1.01}$. And so, $f \in \Omega(g)$.

0.1.h. $f \in \Omega(g)$

0.1.i. $f \in \Omega(g)$

0.1.j. Credit to Raymond Feng for the following justification.
Firstly note that,

$$(\log n)^{\log n} \geq \frac{n}{\log n} \implies (\log n)^{(\log n + 1)} \geq n$$

Taking the logarithm on both sides leads to:

$$(\log n + 1)\log(\log n) \geq \log n$$

And consequently, we have,

$$\log n \log(\log n) \geq \log n \implies \log(\log n) \geq 1$$

0.1.k. $f \in \Omega(g)$

0.1.l. Firstly, note that, $5^{\log_2 n} = n^{\log_2 5}$. Now, since $\log_2 5 > \frac{1}{2}$, this means that $f \in \Omega(g)$.

0.1.m. $f \in \Omega(g)$

0.1.n. $f \in \Theta(g)$

0.1.o. $f \in \Omega(g)$

0.1.p. Firstly, note that,

$$\begin{aligned}
g(n) &= 2^{(\log_2 n)^2} \\
&= 2^{(\log_2 n)(\log_2 n)} \\
&= n^{\log_2 n} \\
f(n) &= (\log n)^{\log n} \\
&= (10^{\log(\log n)})^{\log n} \\
&= (10^{\log n})^{\log(\log n)} \\
&= n^{\log(\log n)}
\end{aligned}$$

Thus, it follows that $f \in \mathcal{O}(g)$.

0.1.q. I could not solve this question. Please contact me if you have the solution.

Exercise 0.2.

0.2.b. If $c = 1$, then the series is just a finite sum of 1's. And so, $g(n) = n + 1 \in \Theta(n)$.

0.2.a. We know that the reciprocals of the powers of any $n > 1$ produce a convergent series, and so, we apply that here. Hence, the infinite series is convergent, which means its partial sums tend to a certain finite limit. And so, for some $c \in \mathbb{R}^+$, we have, $g(n) \leq c \in \Theta(1)$.

0.2.c. We have,

$$\begin{aligned}
g(n) &= \frac{c^{n+1} - 1}{c - 1} \\
&\in \Theta(c^n)
\end{aligned}$$

Exercise 0.3.

0.3.a. We use strong induction to solve this question.

Base case ($n = 6$): $F_6 = 8$. $2^{0.5 \cdot 6} = 2^3 = 8$. This is true.

Inductive hypothesis: Assume that, for all $6 \leq n \leq k$, $F_n \geq 2^{0.5 \cdot n}$.

Inductive step: We have,

$$\begin{aligned}
F_{k+1} &= F_k + F_{k-1} \\
&\geq 2^{0.5\cdot k} + 2^{0.5\cdot(k-1)} && \text{(using induction hypothesis)} \\
&= 2^{0.5\cdot(k-1)} \cdot (1 + 2^{0.5}) \\
&\geq 2^{0.5\cdot(k-1)} \cdot 2 && \text{(since } 2^0.5 \geq 1\text{)} \\
&= 2^{0.5\cdot(k+1)}
\end{aligned}$$

0.3.b. I tried solving this, but I ended up with two upper bounds and could not form a nice inequality. Please contact me if you found the solution.

0.3.c. Same as before.

Exercise 0.4.

0.4.a. Define two 2×2 matrices A and B, with entries a_{ij} and b_{ij} respectively. We have,

$$AB = \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b12 + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b12 + a_{22}b_{22} \end{bmatrix}$$

Well, clearly, this takes 8 multiplications and 4 additions.

0.4.b. Compute X^{2^k} for as long as $2^k \leq n$, and then multiply by the appropriate powers to result in X^n. Each multiplication takes 12 operations which results in approximately $12 \log_2 n$ steps. This is hence, $\mathcal{O}(\log_2 n)$.