Reference for Assignment 2

2023/7/4

Problem 2.1

An algorithm for performing 5 multiplications is given as follows:
$f(x)g(x) = aux^3 + (av + bu)x^2 + [(a + b)(w + v) - av - bw]x + bw$

Feel free to explore other algorithms that require only 4 multiplications.

Problem 2.2

The goal of this problem is not to prove the equivalence of these two cases (in fact, they are not equivalent), but rather to show that for functions $f(n)$ satisfying either of these conditions in the Master theorem, we have $T(n) = \Theta(n^{log_b a})$.

Once this observation is made, the proof of this problem is not difficult. Proof omitted.

Problem 2.3

You can consider functions $f(n)$ that contain non-polynomial and non-monotonic components, such as those involving cos. An example is
$T(n) = T(\frac{n}{2}) + n(1.1 - cos(n))$.

Common incorrect answers include $T(n) = 2T(\frac{n}{2}) + n^2$. However, since $2 * (\frac{n}{2})^2 = \frac{n^2}{2} \leq \frac{2n^2}{3}$, where $\frac{2}{3} < 1$, this satisfies the regularity condition. The regularity condition requires the existence of a constant d less than 1 such that $af(\frac{n}{b}) \leq df(n)$, not for any d less than 1.

Problem 2.4

You can first make a guess or use the method of undetermined coefficients by assuming $T(n) = n^c$ to obtain $T(n) = \mathcal{O}(n^3)$, and then use the substitution method for the proof.

Problem 2.5

$$S_n - 4S_{n-1} = 8(S_{n-1} - 4S_{n-2}) = 8^{n-2}(S_2 - 4S_1) = 8^n$$

$$S_n - 2 * 8^n = 4(S_{n-1} - 2 * 8^{n-1}) = 4^{n-1}(S_1 - 16)$$

$$S_n = 2 * 8^n + (S_1 - 16) * 4^{n-1} = \Theta(8^n)$$

This problem is relatively standard. It was observed during grading that the main issue was failing to provide an asymptotic complexity solution as required by the problem statement.