Homework #1: Divide and Conquer
Due Date: see Canvas

This is "hw1", our first homework. I will also make a related programming challenge via hackerrank, but it will not count as part of your hw1 mark. (I'll count all the programming challenges together, as worth at most one homework for final grading.)

Because of the size of the class, I can only do a cursory job of grading. But do regard these problems as review for the midterm exam. I am willing to give hints and/or time extensions, if you come see me *in person*, preferably during *office hours*.

Assignment

Work out solutions to **SIX** of the following problems, see Canvas for the duedate and submission format. Page and problem/exercise numbers are from the 3rd edition of our textbook. If there is any confusion about which problems I mean, please ask.

Problem 1. Book Search. Find an example of a "divide and conquer" algorithm in our textbook, outside Chapters 4 and 30. It could be in the main text, or in an exercise or problem. Briefly describe the problem, the algorithm, its recurrence, and time bound. Also give Chapter and page numbers. (Bonus: find one that nobody else does!)

Problem 2. Book Problem 4-1, page 107, 7 parts. For each recurrence, indicate which case (if any) of the Master Theorem applies. If the theorem does not apply, give a brief solution instead.

Problem 3. Book Problem 4-3,page 108, 10 parts. For each recurrence, indicate which case (if any) of the Master Theorem applies. If the theorem does not apply, give a brief solution instead.

Problem 4. Book Problem 4-5, Chip pages 109–110, three parts. This is the chip testing problem, give a brief answer for each part.

Problem 5. Book Exercise 30.2-7, page 914. Here you show how to compute the coefficients of a polynomial with n given zeros. I suggest you use divide-and-conquer, with a recurrence, and use the book algorithm which multiplies two polynomials in $O(n \lg n)$ time. (Reminder: when the book says a polynomial has "degree bound $n + 1$", that means its degree is less than $n + 1$.)

Problem 6. Book Problem 30-1, multiplication, pages 920–921, 3 parts. Note parts (b) and (c) use essentially the same time recurrence, but on different problems. For part (c), you may assume we know how to add (or subtract) two B-bit integers in $O(B)$ time.

Problem 7. Example ($n = 4$, complex). Consider the DFT for $n = 4$, with $\omega_4 = i$ (the square root of -1).

7(a). Write out the DFT as a 4x4 matrix V_4. (similar to the top of page 902 or 913). Use explicit complex constants like "i" and "-1" instead of "ω_4^1" and "ω_4^2".

7(b). Write out its inverse matrix $(V_4)^{-1}$. Again, use explicit constants.

7(c). Compute the DFT $\mathbf{y} = V_4\, \mathbf{a}$, for the vector $\mathbf{a} = (1, 2, 3, 4)$. (Note $\mathbf{y}$ and $\mathbf{a}$ are actually column vectors, in this matrix-vector multiplication.)

7(d). Draw the butterfly circuit (like in Figure 30.5 on page 919) for $n = 4$. Label each wire (or node) with the value it holds when applied to the input vector $\mathbf{a}$ given above. (Show the leftmost values for $\mathbf{a}$, the values permuted, the values after the first column of butterfly operations, and the final values ($\mathbf{y}$) after the second column of butterfly operations.

Problem 8. Example ($n = 4$, mod 5). Repeat the previous problem (all parts), but this time doing integer arithmetic modulo 5, with $\omega_4 = 3$. Use the same $\mathbf{a}$ vector. All numbers should be remainders modulo 5 (in the range 0 to 4, inclusive). In particular do not use "1/4", the entries of matrix $(V_4)^{-1}$ should all be integers.