Hi, I'm Kyle!

https://personal.utdallas.edu/~kyle.fox/courses/cs6363.003.21s/

- Learning
- Email
- Homework
- Submit
- Grades

"required book" Cormen et al.

"highly recommended": Erickson
homework: 30% (5 assignments)
two midterms: 20% each
final exam: 30%
cumulative

homework groups of 1 or 2
one submission per group
ask for free 48 hour homework extensions
Can use outside sources if needed, but cite them and write solutions in your own words.
algorithm: an explicit, precise, unambiguous, mechanically-executable sequence of elementary instructions

**BottlesOfBeer**(*n*):
For *i* ← *n* down to 1
   Sing "*i* bottles of beer on the wall, *i* bottles of beer,"
   Sing "Take one down, pass it around, *i* − 1 bottles of beer on the wall."
Sing "No bottles of beer on the wall, no bottles of beer,"
Sing "Go to the store, buy some more, *n* bottles of beer on the wall."
lattice multiplication

input: two arrays $X[0..m-1]$ $Y[0..n-1]$

\[ x = \sum_{i=0}^{m-1} X[i] \cdot 10^i \]

\[ y = \sum_{j=0}^{n-1} Y[j] \cdot 10^j \]

output: $Z[0..m+n-1]$

\[ z = x \cdot y = \sum_{k=0}^{m+n-1} Z[k] \cdot 10^k \]
FIBONACCI_MULTIPLY(X[0..m-1], Y[0..n-1]):
    hold ← 0
    for k ← 0 to n + m - 1
        for all i and j such that i + j = k
            hold ← hold + X[i] · Y[j]
        Z[k] ← hold mod 10
        hold ← ⌊hold/10⌋
    return Z[0..m + n - 1]

Any CS student should be able to run any line without further explanation.
Describing an algorithm:

1) What
   - what precisely does it solve?

2) How
   - describe the algo itself

3) Why
   - prove the algo solves the what

4) How fast: asymptotic run time
remember your audience
(me or a TA)
(not a computer)
(skeptical novice)
nobody should have to read code to know "what"

highly recommend using pseudocode.
use English + math for individual instructions
a proof & restating how in English