Algorithm Analysis

Reagan Shirk

August 27, 2020

Chapter 1 Recap

What is an algorithm?

- Sequence/parallel of simple steps
- A thing that takes input and produces output
- You have deterministic, non-deterministic, randomized, heuristic, and approximation algorithms

Efficiency

- You have time and space efficiency
- Some problems are easy, some are hard (P vs NP, we'll get to that later in the semester)
- You need to analyze the complexity of algorithms to help determine efficiency
- Guys I'm sorry I got distracted by Tinder. This isn't a good sign for the rest of the semester

Chapter 2: Sorting

Insertion Sort

- "You insert to the right place" Thanks Cheng I never would've guessed
- It's basically like sorting a hand of cards
- How do you find the right place? Compare with the previous number
- If you're given an array to sort, you start at the second index (keep in mind that the book indexes at 1, not 0, so the second index is j = 2) and compare it to the previous number. Move it if you need to. Continue with the $j + 1$ index. Continue to compare the index after each move until the number to its left is smaller than it

```
INSERTION-SORT(A, n)
    for j = 2 to n
        key = A[j]
        // Insert A[j] into the sorted sequence A[1...j-1]
        i = j - 1
        while i  > 0 and A[i] > key
            A[i + 1] = A[i]
            i = i - 1
        A[i + 1] = key
```

- Will this program terminate? Yes, it will terminate when `j = n`.
    - Loops are the only things that will cause issues with termination