Algorithm Analysis

Reagan Shirk

September 10, 2020

Asymptotic Bounds

$$\text{A note: } \log^{100}(n) < \sqrt{n}$$
$$n \log^{100} n \stackrel{?}{=} o(n^3) \rightarrow \text{Yes, this is true}$$

Stirling Formula

$$n!\sqrt{2\pi n}\left(\frac{n}{e}\right)^n\left(1+\Theta\left(\frac{1}{n}\right)\right)$$

Things to Note

$$n! \neq \Theta(n^n)$$
$$n! = o(n^n)$$
$$n! = \Theta\left(\sqrt{n}\left(\frac{n}{e}\right)^n\right)$$
$$n! = o\left(2^{n^2}\right)$$
$$\binom{n}{2} = \Theta(n^2)$$
$$\binom{n}{3} = \Theta(n^3)$$
$$3^n = \omega(2^n)$$
$$3^n = \omega(n^{100}2^n)$$

Divide and Conquer (Chapter 4)

- Divide and conquer is when you cut the array down the middle and split the problem into two parts
- Examples of divide and conquer algorithms:
  - Merge sort
    - When there's a tie in a merge sort, which one goes into the output array? The one in the left subarray
  - Binary Search
  - Stock market example
    - You want an algorithm to find optimum buying and selling time
    - You can do an exhaustive search- you will have $\binom{n}{2}$ buying and selling pairs so your complexity will be $\Omega\left(nchoose2\right) = \Omega(n^2)$ which isn't terrible but it could be better. Not efficient
    - You can use a Greedy algorithm: either buy lowest or sell the highest, neither of these are optimal
    - Divide and conquer is the way to go for the most efficient and optimal solution