Intro to Algorithms HW 1

Greg Stewart

January 25, 2018

Q1

Fibonacci numbers are given by

$$F_0 = 0,\ F_1 = 1,\ F_n = F_{n-1} + F_{n-2}$$

(a) *Problem.* Prove $F_n \geq 2^{0.5n}$ for $n \geq 6$.

Proof. For n = 6 (the base case), we have

$$F_6 = 8 \text{ and } 2^{0.5*6} = 2^3 = 8$$

$$F_6 \geq 2^{0.5*6} = 8$$

So the base case is true. Let's assume that this is also true for some $k > 6$, i.e.

$$F_k = F_{k-1} + F_{k-2} \geq 2^{0.5k}$$

and

$$F_{k-1} \geq 2^{0.5(k-1)}$$

Then for the $k+1$ term, we have

$$F_{k+1} = F_k + F_{k-1}$$

So

$$F_{k+1} \geq 2^{0.5k} + 2^{0.5(k-1)}$$

$$F_{k+1} \geq 2^{0.5(k-1)}(2^{0.5} + 1)$$

Since $2^{0.5} + 1 > 2$, we can write that

$$F_{k+1} \geq 2 \cdot 2^{0.5(k-1)} = 2^{0.5k+0.5}$$

So we see that

$$F_{k+1} \geq 2^{0.5(k+1)}$$

(c) *Problem.* Largest c for which $F_n = \Omega(2^{cn})$

We know that

$$F_n = F_{n-1} + F_{n-2}$$

so we can rewrite the inequality as

$$F_n \geq 2^{c(n-1)} + 2^{c(n-2)}$$

$$F_n \geq 2^{cn}(2^{-c} + 2^{-2c})$$

and for the statement to remain true, we need $2^{-c} + 2^{-2c} = 1$.

$$2^{-c} + 2^{-2c} = 1$$

$$2^{-2c} + 2^{-c} - 1 = 0$$

Solving this like a quadratic shows that

$$2^{-c} = -\frac{1 \pm \sqrt{5}}{2}$$

$$c = -\log_2(-\frac{1 \pm \sqrt{5}}{2})$$

Only one of these values is positive, so we must choose that one:

$$c = 0.69424$$

Q2

Let's take a look at some cases:

- $n = 0 \rightarrow 1$ * printed
- $n = 1 \rightarrow 2$ * printed
- $n = 2 \rightarrow 1 * + 2 * + 1 * = 4$ * printed
- $n = 3 \rightarrow 1 * + 1 * + 2 * + 4 * = 8$ * printed

So we have

$$T(0) = 1,\ T(1) = 2,\ T(2) = 4,\ T(3) = 8, \ldots$$

This pattern clearly shows exponential appreciation, and can be written as

$$T(n) = T(n-1) + T(n-2) + T(n-3) + \cdots + T(0)$$

and we know from this that

$$T(n-1) = T(n-2) + T(n-3) + \cdots + T(0)$$

So we can write a formula for $T(n)$:

$$T(n) = 2T(n-1)$$

This formula is equivalent to

$$T(n) = 2^n$$

To show this we use induction.

$$T(0) = 2^0 = 1$$

Assume that $T(k) = 2^k$ for some $k > 0$. Then

$$T(k+1) = 2^{k+1} = 2^k \cdot 2^1 = 2 \cdot 2^k = 2T(k)$$

Which is the same as the formula that we found at first. So it's clear that

$$T(n) = 2^n$$

Q3

(a) $f = n(n+1)$ and $g = 2000n^2$

$$f = \Theta(g)$$

(b) $f = 100n^2$ and $g = 0.01n^3$

$$f = O(g)$$

(c) $f = \log_2 n$ and $g = \ln n$

$$f = O(g)$$

(d) $f = \log_2^2 n$ and $g = \log_2 n^2$

$$f = \Omega(g)$$

(e) $f = 2^{n-1}$ and $g = 2^n$

$$f = \Theta(g)$$

(f) $f = (n-1)!$ and $g = n!$

$$f = O(g)$$

Q4

In order of growth from lowest to highest:

$$5\log_2(n+100)^{10},\ \ln^2 n,\ n^{1/3},\ 0.001n^4 + 3n^3 + 1,\ 2^{2n},\ 3^n,\ (n-2)!$$