Comp 3804 Winter 2017

- Course outline

"Design and Analysis of Algorithms"

- What is an algorithm?

  "Explicit, unambiguous, computational procedure that transforms an input into an output."

  Example: input: ~~map~~ road map, points A and B
  Output: shortest path from A to B.

- Goal: Learn to solve computational problems efficiently.

- When faced with a problem:

  1) How do I approach the problem?
     → Various problem solving techniques.
  2) Is my solution correct?
     → Correctness proofs
  3) Is my solution efficient?
     → Measured by counting "basic operations"
     → ~~Set of~~ Basic operations ~~is~~ are determined by your model of computation

→ Real RAM model counts
  $+, -, \times, \div$, read, write, $<, >, =$
→ Various analysis techniques
→ Limits of efficiency: some problems <u>cannot</u> be solved efficiently

- Mostly pseudocode, very little programming.

<u>Example</u>: Fibonacci Numbers

$F_0 = 0, \quad F_1 = 1, \quad F_n = F_{n-1} + F_{n-2}$ (for $n \geq 2$)

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ...

We can turn this definition into an algorithm

<u>Algorithm</u> fib(n):
  if $n \leq 1$ then return n
  else return fib(n-1) + fib(n-2)

- Correct? Yes.
- Terminates? Yes.
- Efficient?
  "How does the running time depend on the input?"

Define $T(n)$ = #basic operations performed by fib(n).

For ~~0 or~~ $n=0$ or $n=1$: $T(n) = \cancel{2}\underbrace{1}_{\text{"}n\leq 1?\text{"}} + \underbrace{1}_{\text{"return"}}$

For $n \geq 2$: $T(n) = \underbrace{1}_{\text{"}n\leq 1?\text{"}} + \underbrace{1}_{\text{"}n-1\text{"}} + \underbrace{T(n-1)}_{\text{"fib}(n-1)\text{"}} + \underbrace{1}_{\text{"}n-2\text{"}} + \underbrace{T(n-2)}_{\text{"fib}(n-2)\text{"}} + 1 \text{ ~~"return~~} + \underbrace{1}_{\text{"return"}}$

$$T(n) = \begin{cases} 2 & \text{if } n \leq 1 \\ T(n-1) + T(n-2) + 5 & \text{otherwise} \end{cases}$$

$\Rightarrow T(n) \geq F_n$

This is terrible, because F_n is huge:

~~We can show that~~ $F_n \geq 2^{\frac{n-1}{2}}$ for $n \geq 3$

<u>Claim:</u>

<u>Proof:</u> By induction on n.

<u>Base case</u> $n=3$: $F_3 = 2 \cancel{\geq} = 2^1 = 2^{2/2} = 2^{\frac{3-1}{2}}$ ✓

$n=4$: $F_4 = 3 \geq 2^{\frac{4-1}{2}} = 2^{3/2} \approx 2.82\ldots$ ✓

<u>Inductive Hypothesis:</u> Suppose $F_n \geq 2^{\frac{n-1}{2}}$ for all $3 \leq n \leq k$.

<u>Inductive Step:</u> We need to show that $F_{k+1} \geq 2^{\frac{(k+1)-1}{2}}$.

$$\begin{aligned} F_{k+1} &= F_k + F_{k-1} && \text{(by defn.)} \\ &\geq 2^{\frac{k-1}{2}} + 2^{\frac{k-2}{2}} && \text{(by IH)} \\ &\geq 2^{\frac{k-2}{2}} + 2^{\frac{k-2}{2}} \\ &= 2 \cdot 2^{\frac{k-2}{2}} = 2^{1+\frac{k-2}{2}} = 2^{\frac{k}{2}} \end{aligned}$$

□