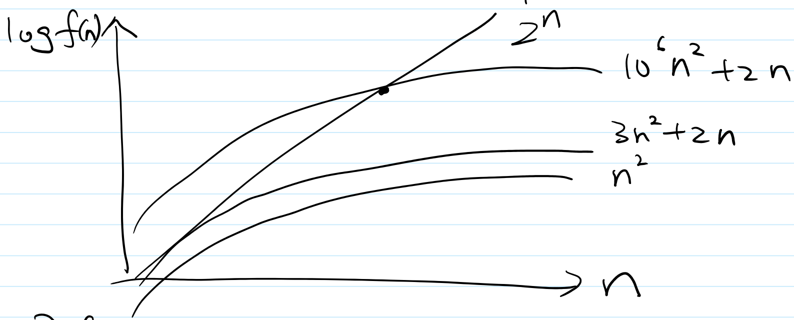
Lecture 1 Asymptotic Notations, Euclid's Algorithm

Tuesday, August 29, 2017 1:06 PM

- Idea: roughly measure the running time / memory for algorithm

$$3n^2+2n \approx n^2 \not\approx 2^n$$



- Definition

"$\leq$" - (1) $f(n) = O(g(n))$, if there is constants $C>0$, $n_0>0$ such that for all $n \geq n_0$, $f(n) \leq C \cdot g(n)$

"$\geq$" (2) $f(n) = \Omega(g(n))$, if there is constants $C>0$, $n_0>0$ such that for all $n \geq n_0$, $f(n) \geq C \cdot g(n)$

"$=$" (3) $f(n) = \Theta(g(n))$ if $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$

"$<$" (4) $f(n) = o(g(n))$ if $g(n) \neq O(f(n))$

"$>$" (5) $f(n) = \omega(g(n))$ if $f(n) \neq O(g(n))$

- Example: (a) $3n^2+2n = O(n^2)$

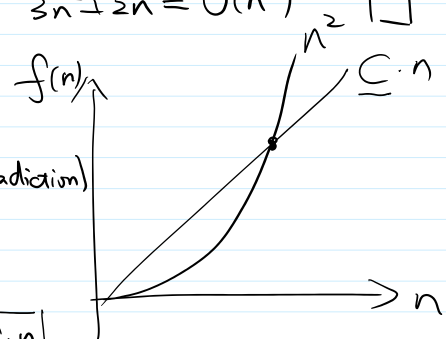
Proof: $3n^2+2n \leq 3n^2+2n^2 = 5n^2$

in the definition, can choose $C=5$, $n_0=1$

$3n^2+2n \leq C \cdot n^2$, so $3n^2+2n = O(n^2)$ $\square$

(b) $n^2 \neq O(n)$

$(n^2 = \omega(n))$



Proof: Assume $n^2 = O(n)$ (towards contradiction)

there is $C>0$, $n_0>0$ s.t.

when $n \geq n_0$, $n^2 \leq C \cdot n$

$$\boxed{\begin{array}{l} \text{want: find a } n \geq n_0 \text{ s.t. } n^2 > C \cdot n \\ \quad n^2 > C \cdot n \\ \Rightarrow \quad n > C \end{array}}$$

pick $n > \max\{n_0, C\}$

then we have $n^2 = n \cdot n > C \cdot n$ contradiction!

therefore $n^2 \neq O(n)$ $\square$

$$\log n < \sqrt{n} < n < n\log n < n^2 < 2^n < 3^n < n!$$

- Bubble Sort

```
for i = n downto 1
    for j = 1 to i-1
        if a[j] > a[j+1] then swap.
```

$$T = (n-1) + (n-2) + \cdots + 1 = \frac{n(n-1)}{2}$$

- what if there is an algorithm that calls Bubble Sort on arrays of size $1, 2, 3, \ldots, n$

$$T = \frac{1\times 0}{2} + \frac{2\times 1}{2} + \boxed{\frac{3\times 2}{2}} + \cdots + \frac{n(n-1)}{2}$$

$$= \frac{\cancel{2\times 1\times 0} - 1\times 0\times(-1)}{6} + \frac{3\times\cancel{2\times 1} - \cancel{2\times 1\times 0}}{6} + \boxed{\frac{4\times\cancel{3\times 2} - 3\times\cancel{2\times 1}}{6}}$$

$$+ \cdots + \frac{(n+1)n(n-1) - \cancel{n(n-1)(n-2)}}{6}$$

$$= \frac{(n+1)n(n-1)}{6}$$

hard

Claim: $T = \Theta(n^3)$

Proof: $T = \sum_{i=1}^{n} \frac{i(i-1)}{2} \le n \cdot \frac{n(n-1)}{2} = O(n^3)$

max

$$T = \sum_{i=1}^{n} \frac{i(i-1)}{2} > \sum_{i=\frac{n}{2}+1}^{n} \frac{i(i-1)}{2} \ge \frac{n}{2} \cdot \frac{\left(\frac{n}{2}\right)^2}{2} = \frac{n^3}{16}$$

$$T(n) = \Omega(n^3)$$

- Euclid's algorithm
  - Goal: compute greatest common divisor (gcd) of 2 integers.

    $\gcd(15, 9) = 3$

  - gcd(a, b)

```
gcd(a, b)
    if b == 0 then
        return a
    else return gcd(b, a mod b)
```

$$\gcd(15,9) \longrightarrow \gcd(9,6) \longrightarrow \gcd(6,3) \longrightarrow \underset{3}{\underline{\gcd(3,0)}}$$

- Proof: By induction

  ① if $b = 0$ $\quad \gcd(a, 0) = a$.

③ induction hypothesis (assume my alg works on small inputs)

assume $\underline{\gcd(a,b) \text{ is correct when } b<n}$ (Know this is true for $n=1$)

want to prove $\gcd(a,b)$ is correct when $b=n$

(my alg also works for larger inputs)

want: $\gcd(a,b) = \gcd(b, a \bmod b)$

Proof: assume $a \bmod b = a - k \cdot b$

① if $\underbrace{c|a}_{c \text{ divides } a}$, $c|b$ then $c \,|\, a \bmod b$

$$\frac{a \bmod b}{c} = \frac{a-kb}{c} = \frac{a}{c} - k \cdot \frac{b}{c} = \text{integer}$$

integers

② if $c|b$, $c \,|\, a \bmod b$ then $c|a$

$$\frac{a}{c} = \frac{(a-kb)+kb}{c} = \frac{a-kb}{c} + k \frac{b}{c} = \text{integer.}$$

integers

①+② ⟹ the set of common divisors for (a,b) and $(b, a \bmod b)$ are the same

$$\underline{\gcd(a,b) = \gcd(b, a \bmod b)}$$

by induction hypothesis, since $a \bmod b < b = n$

$\gcd(b, a \bmod b)$ is computed correctly

therefore $\gcd(a,b)$ is also correct $\square$