Lec 8/31

Thursday, August 31, 2017 12:35

The master theorem

$$f(n) \underset{\text{polynomially smaller.}}{\ll} g(n) \quad \text{iff} \quad f(n) \in O(g(n)\, n^{-\varepsilon}) \text{ or } f(n)\, n^{\varepsilon} \in O(g(n))$$

for some $\varepsilon > 0$.

i.e. $1 \ll \sqrt{n} \ll n^{0.99} \ll n \ll n^2$

$1 \not\ll \log(n)$ and $n \not\ll n\log n$ since $n^{\varepsilon} \notin O(\log n)$

$f(n) \approx g(n)$ iff $f(n) \in \Theta(g(n))$

Note: $\ll$ and $\approx$ are only used in this class.

$$f(n) \ll g(n) \underset{\not\Leftarrow}{\Rightarrow} f(n) \in o(g(n))$$

Master Theorem:

if $T(n) = a\,T(n/b) + f(n)$

1: if $f(n) \ll n^{\log_b(a)}$, then $T(n) \in \Theta(n^{\log_b a})$

2: if $f(n) \gg n^{\log_b(a)}$, then $T(n) \in \Theta(f(n))$

3: if $f(n) \approx n^{(\log_b(a))}$, then $T(n) \in \Theta(f(n) \log(n))$

4: if $f(n) \approx n^{\log_b(a)} \log^k(n)$, then $T(n) \in \Theta(f(n) \log(n))$

↳ for some $k \in \mathbb{N}$

In case 2, it's reqd. that $a f(n/b) \le c f(n)$ for some $c < 1$.

n/b should be interpreted as $\lfloor \frac{n}{b} \rfloor$ or $\lceil \frac{n}{b} \rceil$.

$$T(n) = 3T(\tfrac{n}{4}) + n$$

$$f(n) = n \gg n^{\log_4 3} \Rightarrow T(n) \in \Theta(n)$$

$$T(n) = 9T(\tfrac{n}{3}) + n, \quad n \ll n^2 \Rightarrow T(n) \in \Theta(n^2)$$

$$T(n) = T(\tfrac{2n}{3}) + 1, \quad 1 \approx n^{\log_{3/2} 1} \Rightarrow T(n) \in \Theta(\log n)$$

$$T(n) = 3T(\tfrac{n}{4}) + n\log n \Rightarrow T(n) \in \Theta(n\log n)$$

$$T(n) = 7T(\tfrac{n}{2}) + \Theta(n^2) \Rightarrow T(n) \in \Theta(n^{\log_2 7})$$

$$T(n) = 2T(\tfrac{n}{2}) + n\log n \Rightarrow T(n) \in \Theta(n\log^2 n)$$

$T(n) = T(n/3) + T(2n/3) + n \Rightarrow$ can't use master thm.

$$T(n) = 2T(\sqrt{n}) + \log n$$

Consider n a power of 2, $n = 2^m$

$$T(2^m) = 2T(2^{m/2}) + \log(n)$$

let $S(m) = T(2^m) = T(n)$

$$S(m) = 2S(m/2) + m$$

$S(m) \in \Theta(m \log m)$

$S(m) = T(n) \in \Theta(\log(n) \log(\log(n)))$ for $n = 2^m$

and since $\log(n) \log(\log(n))$ is smooth,

$T(n) \in \Theta(\log(n) \log(\log(n)))$

Strassen's Algorithm for Matrix Multiplication

$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$ algorithm requires $\Theta(n^3)$ time.

$$C = AB$$

$$\underset{C}{\begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}} = \underset{A}{\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}} \underset{B}{\begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}}$$

$$C_{ij} = A_{i1} B_{1j} + A_{i2} B_{2j}$$

(not Strassen's, still $\Theta(n^3)$)

on n^3 processors:

$$T(n) = 1\, T\left(\frac{n}{2}\right) + 1 \quad \in \Theta(\log(n))$$