COMP3121/9101 Week02

Asymptotic notation

- "Big Oh"
    - f(n) = O(g(n)) is an abbreviation for: “There exist **positive** constants c and n0 such that 0f(n)cg(n) for all **nn0**”.
    -
    - f(n) = O(g(n)) means that f(n) does not grow substantially faster than g(n) because a multiple of g(n) eventually dominates f(n).
- "Omega"
    - f(n) = (g(n)) is an abbreviation for: “There exists **positive** constants c and n0 such that 0cg(n)f(n) for all **nn0**.”
    - g(n)=O(f(n))
    -
    - f(n) = Ω(g(n)) essentially says that f(n) grows at least as fast as g(n), because f(n) eventually dominates a multiple of g(n).
- "Theta"
    - f(n) = (g(n)) if and only if **f(n) = O(g(n))** and **f(n) = (g(n))**; thus, f(n) and g(n) have the same asymptotic growth rate.
    -

Recurrences

Let a 1 be an integer and b > 1 a real number; Assume that a **divide-and-conquer algorithm**:

reduces a problem of size n to a many problems of smaller size **n/b**; the overhead cost of splitting up/combining the solutions for size n/b into a solution for size n is if **f(n)**,

then the time complexity of such algorithm satisfies

$$T(n) = aT(ceiling(\frac{n}{b})) + f(n)T(n) = aT(\frac{n}{b}) + f(n)$$

- recurrence
-

Master Theorem

Let:

- a 1 be an integer and and b > 1 a real;
- f(n) > 0 be a non-decreasing function;
- T (n) be the solution of the recurrence T (n) = a T (n/b) + f (n);

Then:

1. If $f(n) = O(n^{\log_b a})$ for some > 0, then $T(n) = (n^{log_b a})$;
2. If $f(n) = (n^{\log_b a})$, then $T(n) = (n^{\log_b a} \log_2 n)$;
3. If $f(n) = (n^{\log_b a+})$ for some > 0, and for some **c<1** and some n_0,

   $af(n/b)cf(n)$ holds for all $n > n_0$, then $T(n) = (f(n))$;
4. If none of these conditions hold, the Master Theorem is NOT applicable.

Examples

$T(n) = 3T(\frac{n}{2}) + n(2 + \cos n)$

a=3

b = 2

f(n)= $n(2 + \cos n)$ [n, 3n] = O(n)

$n^{\log_b a} = n^{\log_2 3} \approx n^{1.58} > n$

2+cos n [1, 3]

f(n)=$O(n^{1.58-\epsilon})$ for ϵ <0.5

T(n)=$\Theta(n^{\log_2 3})$

Solution

For the given equation $T(n) = 3T(\frac{n}{2}) + n(2 + cosn)$,

$a = 3$, $b = 2$, $f(n) = n(2 + cosn) = O(n)$ because $cosn$ is bounded at $[-1, 1]$, $2 + \cos n$ is a positive constant.

so we have $a \geq 1, b > 1$, $n^{log_b a} = n^{log_2 3} \approx n^{1.58}$, then $f(n) = O(n^{log_2 3-\epsilon})$, for some $0 < \epsilon < 0.4$, which is case 1 of Master Theorem, then $T(n) = \Theta(n^{log_2 3})$.

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

a = 3, b = 4

$n^{log_b a} = n^{log_4 3} \approx n^{0.79} < n^1 = n$

f(n)=$n^{4/3}$

$f(n) = \Omega(n^{log_4 3+\epsilon})$ epsilon < 0.3

3f (n/4) cf(n)

$3(n/4)^{4/3} \leq cn^{4/3}$

c > 3/4^{4/3}=0.47...

Solution

For the given equation $T(n) = 3T(\frac{n}{4}) + n^{\frac{4}{3}}$,

$a = 3$, b =4, $f(n) = n^{\frac{4}{3}}$,

so we have $a \geq 1, b > 1$, $n^{log_b a} = n^{log_4 3} \approx n^{0.78}$, then $f(n) = \Omega(n^{log_3 4+\epsilon})$, for some $0 < \epsilon < 0.07$, there exists some constant $c > 0$ such that $af(n/b) = cf(n)$, which is the third case of Master Theorem, then $T(n) = \Theta(f(n)) = \Theta(n^{\frac{4}{3}})$.

$T(n) = 5T(\frac{n}{2}) + n^{log_2 5}(1 + sin(\frac{2\pi n}{3}))$

Solution

For the given equation $T(n) = 5T(\frac{n}{2}) + n^{log_2 5}(1 + sin(\frac{2\pi n}{3}))$,

$a = 5$, $b = 2$, $f(n) = n^{log_2 5}(1 + sin(\frac{2\pi n}{3}))$,

so we have $a \geq 1, b > 1$, let $g(n) = n^{log_b a} = n^{log_2 5}$, then we should have $f(n) = \Theta(g(n))$to apply master theorem.

To show, $f(n) = \Theta(g(n))$ there exist constant c_1, c_2 and n_0 such that

$0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)$ for all $n \geq n_0$.

We know that n is an integer, then $sin(\frac{2\pi n}{3}))$ is bounded at $[-\frac{\sqrt{3}}{2}, \frac{\sqrt{3}}{2}]$, hence $(1 + sin(\frac{2\pi n}{3}))$ is bounded at $[1 - \frac{\sqrt{3}}{2}, 1 + \frac{\sqrt{3}}{2}]$. Thus we can take $0 < c_1 \leq 1 - \frac{\sqrt{3}}{2}$ and $c_2 \geq 1 + \frac{\sqrt{3}}{2}$, the equation holds for all $n > n_0$. By master theorem, $T(n) = \Theta(n^{log_2 5} lgn)$.

$T(n) = T(n1) + log(n)$,

Solution

For the given equation $T(n) = T(n1) + log(n)$,

$a = 1$, $b = 1$, $f(n) = log(n)$, which does not satisfy the condition of Master Theorem.

$T(n) = T(n-1) + log(n)$
$= T(n-2) + log(n-1) + log(n)$
$= T(n-3) + log(n-2) + log(n-1) + log(n)$
$= T(0) + log(1) + ... + log(n)$
$= T(0) + log(1 \times 2 \times 3 \times ... \times n)$
$= T(0) + log(n!)$
$\leq T(0) + log(n^n)$
$= O(nlogn)$

Prove Master Theorem (Slides)

T(n) = 2 T(n/2) + n log n

Master Therom does not apply

Karatsuba trick analysis

$T(n) = 3(\frac{T}{2}) + cn$

$b = 2; f(n) = cn; n^{\log_b a} = n^{\log_2 3}$

a=3;
since $1.5 < \log_2 3 < 1.6$ we have

$f(n) = cn = O(n^{\log_2 3})$ for any 0 << 0.5

Thus, the first case of the Master Theorem applies. Consequently,

$T(n) = (n^{\log_2 3}) < (n^{1.585})$

Dividing into 3 pieces

lecture slides)

$T(n) = 5T(\frac{n}{3}) + cn$

(b) T (n) = 2T (n/2) + n + log n

a=2

b=2

n$^{\{\text{log_22}\}=\text{n}}$1

f(n)=nˆ(1/2)+log(n)

$f(n) = O(n^{1-\epsilon})\epsilon < 0.3$

Master theorem case 1

Slice into p+1 many slices

The general case - slicing the input numbers A, B into p + 1 many slices

Linear Convolutiom

DFT

(tut questions)