CS F364: Design & Analysis of Algorithm

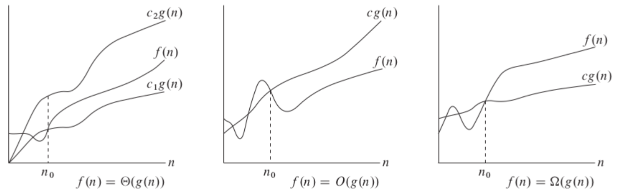
02 Master Method and Integer Multiplication

Dr. Kamlesh Tiwari
Assistant Professor, Department of CSIS,
BITS Pilani, Pilani Campus, Rajasthan-333031 INDIA

Jan 20, 2021 ONLINE (Campus @ BITS-Pilani Jan-May 2021)

http://ktiwari.in/algo

Asymptotic Notation $\Theta, O, o, \omega, \Omega$; zoo



Θ

$\Theta(g(n)) = \{f(n)$: there exists positive constants c_1, c_2 and n_0 such that $0 \le c_1 g(n) \le f(n) \le c_2 g(n)$ for all $n \ge n_0\}$

- $\Theta(x) = \{3x, 5x + 4, ...\}$
- $\Theta(\log x) = \{4 \log x, 5 \log(x^3), 5 \log(x^3) + 2, ...\}$
- We write $5x + 4 = \Theta(x)$ to mean $5x + 4 \in \Theta(x)$

Asymptotic Notation O

Θ

$\Theta(g(n)) = \{f(n)$: there exists positive constants c_1, c_2 and n_0 such that $0 \le c_1 g(n) \le f(n) \le c_2 g(n)$ for all $n \ge n_0\}$

O

$O(g(n)) = \{f(n)$: there exists positive constants c and n_0 such that $0 \le f(n) \le cg(n)$ for all $n \ge n_0\}$

- $O(x) = \{3x, 5x + 4, 3\sqrt{x} + 4\ , 3\sqrt{x} + 4 \log x\ , 7, ...\}$

Asymptotic Notation o

O

$O(g(n)) = \{f(n)$: there exists positive constants c and n_0 such that $0 \le f(n) \le cg(n)$ for all $n \ge n_0\}$

o

$o(g(n)) = \{f(n)$: for any positive constants c, there exists a constant $n_0 > 0$ such that $0 \le f(n) < cg(n)$ for all $n \ge n_0\}$

- $O(x) = \{3x, 5x + 4, 3\sqrt{x} + 4, 3\sqrt{x} + 4 \log x, 7, ...\}$
- $o(x) = \{3\sqrt{x} + 4, 3\sqrt{x} + 4 \log x, 7, ...\}$

$$\lim_{n \to \infty} f(n)/g(n) = 0$$

Asymptotic Notation Ω

Θ

$\Theta(g(n)) = \{f(n)$: there exists positive constants c_1, c_2 and n_0 such that $0 \le c_1 g(n) \le f(n) \le c_2 g(n)$ for all $n \ge n_0\}$

Ω

$\Omega(g(n)) = \{f(n)$: there exists positive constants c and n_0 such that $0 \le cg(n) \le f(n)$ for all $n \ge n_0\}$

- $\Omega(x) = \{3x, 5x + 4, 3x\sqrt{x} + 4\ , 3x\sqrt{x} + 4 \log x\ , 5x^6 + 3x^4 + 5, ...\}$

Asymptotic Notation ω

Ω

$\Omega(g(n)) = \{f(n)$: there exists positive constants c and n_0 such that $0 \le cg(n) \le f(n)$ for all $n \ge n_0\}$

ω

$\omega(g(n)) = \{f(n)$: for any positive constants c, there exists a constant $n_0 > 0$ such that $0 \le cg(n) < f(n)$ for all $n \ge n_0\}$

- $\Omega(x) = \{3x, 5x + 4, 3x\sqrt{x} + 4, 3x\sqrt{x} + 4 \log x, 5x^6 + 3x^4 + 5, ...\}$
- $\omega(x) = \{3x\sqrt{x} + 4\ , 3x\sqrt{x} + 4 \log x\ , 5x^6 + 3x^4 + 5, ...\}$

$$\lim_{n \to \infty} f(n)/g(n) = \infty$$

Recurrence Relation

Equations of the form

$$T(n) = \begin{cases} \Theta(1) & \text{if } x \le c \\ aT(n/b) + f(n) & \text{otherwise} \end{cases}$$

How to solve?

1. Substitution: guess the solution and test
2. Iteration: convert into summation and apply bounds
3. Master method

Substitution

Consider equation

$T(n) = 2T(\lfloor n/2 \rfloor) + n$

Let we guess the solution to be $T(n) = O(n \log n)$

$$\begin{aligned} T(n) &\le 2(c\lfloor n/2 \rfloor \log(\lfloor n/2 \rfloor)) + n & (1)\\ &\le cn\log(n/2) + n & (2)\\ &= cn\log(n) - cn\log 2 + n & (3)\\ &= cn\log(n) - cn + n & (4)\\ &\le cn\log(n) & (5) \end{aligned}$$

As long as $c > 1$

Iteration

Consider equation

$T(n) = 3T(\lfloor n/4 \rfloor) + n$

$$\begin{aligned} T(n) &= n + 3T(\lfloor n/4 \rfloor) & (6)\\ &= n + 3(\lfloor n/4 \rfloor + 3T(\lfloor n/16 \rfloor)) & (7)\\ &= n + 3(\lfloor n/4 \rfloor + 3(\lfloor n/16 \rfloor + 3T(\lfloor n/64 \rfloor)) & (8)\\ &= n\sum_{i=0}^{\infty}(3/4)^i + \Theta(3^{\log_4 n}) & (9)\\ &= 4n + o(n) & (10)\\ &= O(n) & (11)\\ & & (12) \end{aligned}$$

Iteration stops when $\lfloor n/4^i \rfloor = 1$ that is $i = \log_4 n$

Master method

When $T(n) = aT(n/b) + f(n)$ $\quad a \ge 1, b > 1 \quad$ n is positive

Let $\epsilon > 0$ be a constant

1. If $f(n) = O(n^{log_b a - \epsilon})$ then $T(n) = \Theta(n^{log_b a})$
2. If $f(n) = \Theta(n^{log_b a})$ then $T(n) = \Theta(n^{log_b a} \log n)$
3. If $f(n) = \Omega(n^{log_b a + \epsilon})$ then $T(n) = \Theta(f(n))$ provided if $af(n/b) \le cf(n)$ for some constant $c < 1$ and all sufficiently large n. **Regularity condition must be checked in case-3.**

Integer Multiplication

- How do you multiply integers? How much time it takes?
- If $x = x_1 \times 10^{n/2} + x_0$
- Then $xy = x_1y_1.10^n + (x_1y_0 + x_0y_1).10^{n/2} + x_0y_0$

$$T(n) \le 4T(n/2) + c.n$$

Integer Multiplication

Recursive Multiply parts

Algorithm 1: Rec-Mul (x,y)

```
1 p = Rec-Mul (x1 + x0, y1 + y0)
2 x1y1 = Rec-Mul (x1, y1)
3 x0y0 = Rec-Mul (x0, y0)
4 return x1y1 × 10^n + (p − x1y1 − x0y0) × 10^(n/2) + x0y0
```

Time complexity

$T(n) \le 3T(n/2) + c.n$

$O(n^{\log_2 3}) = O(n^{1.59})$

Thank You!

Thank you very much for your attention! (Reference[1])

Queries ?

[1] [1] Book - *Algorithm Design*, Kleinberg Tardos