Notations

Kuan-Yu Chen (陳冠宇)

2019/03/06 @ TR-310-1, NTUST

Review

- Asymptotic notations

Definition [*Big "oh"*]: $f(n) = O(g(n))$ (read as "f of n is big oh of g of n") iff (if and only if) there exist positive constants c and n_0 such that $f(n) \le cg(n)$ for all n, $n \ge n_0$. □

Definition: [Omega] $f(n) = \Omega(g(n))$ (read as "f of n is omega of g of n") iff there exist positive constants c and n_0 such that $f(n) \ge cg(n)$ for all n, $n \ge n_0$. □

Definition: [Theta] $f(n) = \Theta(g(n))$ (read as "f of n is theta of g of n") iff there exist positive constants c_1, c_2, and n_0 such that $c_1 g(n) \le f(n) \le c_2 g(n)$ for all n, $n \ge n_0$. □

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

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

Properties.

- Transitivity

$$
\begin{aligned}
f(n) &= \Theta(g(n)) \text{ and } g(n) = \Theta(h(n)) \text{ imply } f(n) = \Theta(h(n)) \\
f(n) &= O(g(n)) \text{ and } g(n) = O(h(n)) \text{ imply } f(n) = O(h(n)) \\
f(n) &= \Omega(g(n)) \text{ and } g(n) = \Omega(h(n)) \text{ imply } f(n) = \Omega(h(n)) \\
f(n) &= o(g(n)) \text{ and } g(n) = o(h(n)) \text{ imply } f(n) = o(h(n)) \\
f(n) &= \omega(g(n)) \text{ and } g(n) = \omega(h(n)) \text{ imply } f(n) = \omega(h(n))
\end{aligned}
$$

Properties..

- Reflexivity

$$
\begin{aligned}
f(n) &= \Theta(f(n)) \\
f(n) &= O(f(n)) \\
f(n) &= \Omega(f(n))
\end{aligned}
$$

- Symmetry

$$f(n) = \Theta(g(n)) \text{ if and only if } g(n) = \Theta(f(n))$$

- Transpose symmetry

$$
\begin{aligned}
f(n) = O(g(n)) &\text{ if and only if } g(n) = \Omega(f(n)) \\
f(n) = o(g(n)) &\text{ if and only if } g(n) = \omega(f(n))
\end{aligned}
$$

Properties...

- Reflexivity

$$
\begin{aligned}
f(n) &= \Theta(f(n)) \\
f(n) &= O(f(n)) \\
f(n) &= \Omega(f(n))
\end{aligned}
$$

$$
\begin{array}{lll}
f(n) = O(g(n)) & \text{is like} & a \le b\ , \\
f(n) = \Omega(g(n)) & \text{is like} & a \ge b\ , \\
f(n) = \Theta(g(n)) & \text{is like} & a = b\ , \\
f(n) = o(g(n)) & \text{is like} & a < b\ , \\
f(n) = \omega(g(n)) & \text{is like} & a > b\ .
\end{array}
$$

- Symmetry

$$f(n) = \Theta(g(n)) \text{ if and only if } g(n) = \Theta(f(n))$$

- Transpose symmetry

$$
\begin{aligned}
f(n) = O(g(n)) &\text{ if and only if } g(n) = \Omega(f(n)) \\
f(n) = o(g(n)) &\text{ if and only if } g(n) = \omega(f(n))
\end{aligned}
$$

Standard Notations.

- Monotonicity
  - A function $f(n)$ is ***monotonically increasing*** if $m \leq n$ implies $f(m) \leq f(n)$
  - It is ***monotonically decreasing*** if $m \leq n$ implies $f(m) \geq f(n)$
  - A function $f(n)$ is ***strictly increasing*** if $m < n$ implies $f(m) < f(n)$
  - A function $f(n)$ is ***strictly decreasing*** if $m < n$ implies $f(m) > f(n)$

Standard Notations..

- Floors and Ceilings
  - For any **real number** x
    - We denote the **greatest integer** less than or equal to x by $\lfloor x \rfloor$

      The floor of x

      Example: $x = 2.7, \lfloor x \rfloor = \lfloor 2.7 \rfloor = 2$

    - We denote the **least integer** greater than or equal to x by $\lceil x \rceil$

      The ceiling of x

      Example: $x = 2.7, \lceil x \rceil = \lceil 2.7 \rceil = 3$

    - For all real x

$$x - 1 < \lfloor x \rfloor \leq x \leq \lceil x \rceil < x + 1$$

    - For any integer x

$$\left\lfloor \frac{x}{2} \right\rfloor + \left\lceil \frac{x}{2} \right\rceil = x$$

Standard Notations…

- For any real number $x \geq 0$ and integers $a, b > 0$

$$\left\lceil \frac{\left\lceil \frac{x}{a} \right\rceil}{b} \right\rceil = \left\lceil \frac{x}{ab} \right\rceil$$

$$\left\lfloor \frac{\left\lfloor \frac{x}{a} \right\rfloor}{b} \right\rfloor = \left\lfloor \frac{x}{ab} \right\rfloor$$

$$\left\lceil \frac{a}{b} \right\rceil \leq \frac{a + (b - 1)}{b}$$

$$\left\lfloor \frac{a}{b} \right\rfloor \geq \frac{a - (b - 1)}{b}$$

Appendix.

- For any real number $x \geq 0$ and integers $a, b > 0$, we want to proof that $\left\lceil \frac{\lceil \frac{x}{a} \rceil}{b} \right\rceil = \left\lceil \frac{x}{ab} \right\rceil$
  - Let $k = \left\lceil \frac{x}{a} \right\rceil$
    - We can get $k - 1 < \frac{x}{a} \leq k \Rightarrow \frac{k-1}{b} < \frac{x}{ab} \leq \frac{k}{b}$
    - Consequently, $\left\lceil \frac{x}{ab} \right\rceil \leq \left\lceil \frac{k}{b} \right\rceil$
  - Suppose that $\left\lceil \frac{x}{ab} \right\rceil < \left\lceil \frac{k}{b} \right\rceil$, then there must be an integer l
    - $\left\lceil \frac{x}{ab} \right\rceil \leq l < \left\lceil \frac{k}{b} \right\rceil \Rightarrow \frac{x}{ab} \leq l < \frac{k}{b} \Rightarrow \frac{x}{a} \leq bl < k \Rightarrow \left\lceil \frac{x}{a} \right\rceil \leq bl < k$
    - So, $k = \left\lceil \frac{x}{a} \right\rceil \leq bl < k \rightarrow\leftarrow$
  - Thus, $\left\lceil \frac{x}{ab} \right\rceil = \left\lceil \frac{k}{b} \right\rceil = \left\lceil \frac{\lceil \frac{x}{a} \rceil}{b} \right\rceil$

Appendix..

- For any real number $x \geq 0$ and integers $a, b > 0$, we want to proof that $\left\lceil \frac{a}{b} \right\rceil \leq \frac{a+(b-1)}{b}$
  - Let $a = kb + r$
    - $\left\lceil \frac{a}{b} \right\rceil = \left\lceil \frac{kb+r}{b} \right\rceil$
    - $\frac{a+(b-1)}{b} = \frac{kb+r+b-1}{b} = \frac{(k+1)b+(r-1)}{b}$
  - If $r = 0$
    - $\left\lceil \frac{a}{b} \right\rceil = \left\lceil \frac{kb+r}{b} \right\rceil = \left\lceil \frac{kb}{b} \right\rceil = k$
    - $\frac{a+(b-1)}{b} = \frac{(k+1)b+(r-1)}{b} = k + 1 - \frac{1}{b} = k + \left(1 - \frac{1}{b}\right)$
    - $\because \left(1 - \frac{1}{b}\right) \geq 0$
    - $\therefore \left\lceil \frac{a}{b} \right\rceil = k \leq k + \left(1 - \frac{1}{b}\right) = \frac{a+(b-1)}{b}$

Appendix...

- For any real number $x \geq 0$ and integers $a, b > 0$, we want to proof that $\left\lceil \frac{a}{b} \right\rceil \leq \frac{a+(b-1)}{b}$
  - Let $a = kb + r$
    - $\left\lceil \frac{a}{b} \right\rceil = \left\lceil \frac{kb+r}{b} \right\rceil$
    - $\frac{a+(b-1)}{b} = \frac{kb+r+b-1}{b} = \frac{(k+1)b+(r-1)}{b}$
  - If $1 \leq r < b$
    - $\left\lceil \frac{a}{b} \right\rceil = \left\lceil \frac{kb+r}{b} \right\rceil = k + 1$
    - $\frac{a+(b-1)}{b} = \frac{(k+1)b+(r-1)}{b} = (k+1) + \frac{r-1}{b}$
    - $\because \frac{r-1}{b} \geq 0$
    - $\therefore \left\lceil \frac{a}{b} \right\rceil = k + 1 \leq (k+1) + \frac{r-1}{b} = \frac{a+(b-1)}{b}$

Standard Notations….

- Modular Arithmetic
  - For any integer a and any positive integer n, the value $a \bmod n$ is the ***remainder*** (or ***residue***) of the quotient $\frac{a}{n}$

$$a \bmod n = a - n\left\lfloor \frac{a}{n} \right\rfloor$$

$$0 \leq a \bmod n < n$$

  - If $(a \bmod n) = (b \bmod n)$
    - $a \equiv b \pmod{n}$
    - a is ***equivalent*** to b, modulo n

Standard Notations.....

- Polynomials
  - Given a nonnegative integer d, a ***polynomial in*** n ***of degree*** d is a function $p(n)$ of the form

$$p(n) = \sum_{i=0}^{d} a_i n^i$$

    - $a_0, a_1, \ldots, a_d$ are the coefficients
    - $a_d \neq 0$

  - We say that a function $f(n)$ is ***polynomially bounded*** if $f(n) = O(n^k)$ for some constant k

Standard Notations……

- Exponentials
  - For all real $a > 0$, m, and n, we have the following identities

$$
\begin{aligned}
a^0 &= 1 , \\
a^1 &= a , \\
a^{-1} &= 1/a , \\
(a^m)^n &= a^{mn} , \\
(a^m)^n &= (a^n)^m , \\
a^m a^n &= a^{m+n} .
\end{aligned}
$$

Standard Notations……

- Logarithms
  - We shall use the following notations:

$$\begin{aligned} \lg n &= \log_2 n && \text{(binary logarithm)} , \\ \ln n &= \log_e n && \text{(natural logarithm)} , \\ \lg^k n &= (\lg n)^k && \text{(exponentiation)} , \\ \lg \lg n &= \lg(\lg n) && \text{(composition)} . \end{aligned}$$

  - For all real $a > 0$, $b > 0$, $c > 0$, and n,

$$\begin{aligned} a &= b^{\log_b a} , \\ \log_c(ab) &= \log_c a + \log_c b , \\ \log_b a^n &= n \log_b a , \\ \log_b a &= \frac{\log_c a}{\log_c b} , \\ \log_b(1/a) &= -\log_b a , \\ \log_b a &= \frac{1}{\log_a b} , \\ a^{\log_b c} &= c^{\log_b a} , \end{aligned}$$

Recursive Algorithms.

- The recursive mechanisms are extremely powerful, because they often can express a complex process very clearly

- Recursive functions can be categorized into three classes
  - Direct Recursion
    - The function may call itself before it is done
  - Indirect Recursion
    - The function may call other functions that again invoke the calling function
  - Tail Recursion
    - The function may call itself at the end of the function
    - A special case of direct recursion

Recursive Algorithms..

Direct Recursion

```
function A()
{
    ...

    A() ;

    ...
}
```

Indirect Recursion

```
function A()
{
    ...

    B() ;

    ...
}

function B()
{
    ...

    A() ;

    ...
}
```

calling cycle

Tail Recursion

```
function A()
{
    ...

    A() ;
}
```

Recursive Algorithms…

- Let's make a comparison

Recursion	Non-Recursion
Codes are more compact	Codes are complicated
Easy to understand	Hard to read
Time-consuming	Time-saving

Questions?

kychen@mail.ntust.edu.tw