Tutorial 9: The Assignment 4

Qu Xiaofeng, Teaching Asistant

COMP210
Discrete Structure

November 11, 2011

THE HONG KONG POLYTECHNIC UNIVERSITY
香港理工大學

Table of Contents

Review of Algorithm & Counting methods

Pseudo Code of Algorithms

- Algorithms are recipes to solve problems
  - Finite, precise
- For, while, if … then …, if … else if … then …
- Recursive algorithms
  - A routine that calls itself (with a reduced input)

Review of Algorithm & Counting methods cont.

Algorithmic Complexity

- Measures the # of basic operations
  - A function of input size
- Asymptotic notation (Big-O, Big-Ω , Big-Θ, small-o, small-ω)
  - Definitions
  - Finding the dominating terms
  - Write functions in forms of the asymptotic notations and compare their complexity

Review of Algorithm & Counting methods cont.

Big-O definition

DEF: Let f, g be functions with domain $\mathbf{R}_{\geq 0}$ or $\mathbf{N}$ and codomain $\mathbf{R}$. If there are constants C and k such

$$\forall x > k, |f(x)| \leq C \cdot |g(x)|$$

then we write:

$$f(x) = O(g(x))$$

- Big-Θ: f(x) = O(g(x)) & g(x) = O(f(x))

Review of Algorithm & Counting methods cont.

Rule of thumbs

- First, for input size n, determine the # of basic operations as f(n)
- Find the dominating term in f(n)
- The following functions are in growing order of complexity

$$\frac{1}{x}, \ln x, x, x^e, e^x, x^x$$

Review of Algorithm & Counting methods cont.

Counting methods

- Multiplication principle
  - Count in stages
- Addition principle
  - Divide the original set into disjoint sets
- Inclusion-exclusion principle
  - Generalization of the addition principle to overlapping sets
- Pigeon hole principle
  - Given N pigeon, k holes, at least one hole contains $\lceil N/k \rceil$ pigeons
  - Can also solve the inverse problem, how big N needs to be such that for k holes, at least one hole contains $\lceil N/k \rceil$ pigeons

Problem 1

Write an algorithm that reverses a string $s_1, \ldots, s_n$.
Example: If the sequence is AMY BRUNO ELIE,
the reversed sequence is ELIE BRUNO AMY.

Problem 1 cont.

Algorithm 1 Reverse string s

```
Input: String s, where s ends with EOL
i ← 1, word_cnt ← 0 {Parsing}
while s(i) ≠ EOL do
    while s(i) =" ␣ " do
        i ← i + 1
    end while
    if s(i) ≠ EOL then
        word_cnt ← word_cnt + 1, char_cnt ← 1
        while s(i) ≠ EOL and s(i) =" ␣ " do
            word[word_cnt][char_cnt] ← s(i)
            char_cnt ← char_cnt + 1, i ← i + 1
        end while
    end if
end while
for i ← word_cnt to 1 do {Output}
    print word[i]
end for
```

Common Growth Functions

Table: Common Growth Functions (Table 4.3.3)

Theta Form	Name
$\Theta(1)$	Constant
$\Theta(\lg \lg n)$	Log log
$\Theta(\lg n)$	Log
$\Theta(n)$	Linear
$\Theta(n \lg n)$	n log n
$\Theta(n_2)$	Quadratic
$\Theta(n_3)$	Cubic
$\Theta(n_k)$	Polynomial
$\Theta(c_n)$	Exponential
$\Theta(n!)$	Factorial

Problem 4.1

Select a theta notation from Table 4.3.3 for $3n^2 + 2n\lg n$.

$$0 \le \lg n \le n \qquad \textit{for all } n \ge 1$$

$$0 \le 2n\lg n \le 2n^2 \qquad \textit{for all } n \ge 1$$

Then the dominating term is $3n^2$.

$$f(n) = 3n^2 + 2n\lg n \ge 3n^2 = C_1 n^2,\ \textit{where } C_1 = 3$$

$$f(n) = O(n^2)$$

$$f(n) = 3n^2 + 2n\lg n \le 3n^2 + 2n^2 = C_2 n^2,\ \textit{where } C_2 = 5$$

$$f(n) = \Omega(n^2)$$

$$f(n) = \Theta(n^2) \quad \square$$

Problem 4.3

Select a theta notation from Table 4.3.3 for $\frac{(n+1)(n+3)}{n+1}$

For all $n > -1$, the equation could be simplified as bellow,

$$\frac{(n+1)(n+3)}{n+1} = n+3$$

So for all $n \geq 3$,

$$f(n) = \frac{(n+1)(n+3)}{n+1} \geq n = C_1 n = O(n)$$
$$f(n) \leq 2n = C_2 n = \Omega(n)$$

Then

$$f(n) = \frac{(n+1)(n+3)}{n+1} = \Theta(n) \quad \square$$

Problem 5.1

Express in theta notation the number of times the statement x = x + 1 is executed.

```
for i = 1 to n
    for j = 1 to n
        x = x + 1;
```

Problem 5.1 cont.

The basic operation runs 1 times. The for loops of j, runs n times, and the outer for loops of i runs n times. So based on the multiplication principle. Then total number is $1 \times n \times n = n^2$.

$$f(n) = n^2 = \Theta(n^2) \quad \square$$

Problem 6

show that $\lg(n^k + c) = \Theta(\lg n)$ *for every fixed* $k > 0$ *and* $c > 0$.

$$
\begin{aligned}
&\text{for all } n \geq \lg \frac{c}{k}, \\
&\qquad \lg\left(n^k + c\right) \leq \lg(2n^k) \\
&\qquad\qquad = k \lg n + \lg 2 \\
&\qquad\qquad \leq C_1 \lg n, \ \textit{where } C_1 = k + 1, \ n \geq 2 \\
&\qquad\qquad = \Omega(\lg n) \\
&\qquad \lg\left(n^k + c\right) \geq \lg(n^k) = k \lg n \\
&\qquad\qquad = C_2 \lg n, \ \textit{where } C_2 = k \\
&\qquad\qquad = O(\lg n) \\
&\qquad \lg\left(n^k + c\right) = \Theta(\lg n) \quad \square
\end{aligned}
$$

Problem 10

Two dice are rolled, one blue and one red. How many outcomes give the sum of 2 or the sum 12?

Table: Outcomes of dice

Sum	Blue	Red
2	1	1
12	6	6
8	2	6
	3	5
	4	4
	5	3
	6	2

1 outcome gives the sum of 2;
1 outcome gives the sum of 12. □

Problem 12.2

For integers from 5 to 200, inclusive. How many do not contain the digit 0?

Single digit	5,6,7,8,9	5
Two digit (xx)	9×9	81
Three digit (1xx)	9×9	81

By Addition principle, the total is 167. □

Problem 14.1

How many symmetric and antisymmetric relations are there on an n-element set?

Definition

A relation R on a set X is symmetric if $\forall$ x,y $\in$ X, if (x,y) $\in$ R, then (y,x) $\in$ R.{Definition 3.3.9}

Definition

A relation R on a set X is antisymmetric if $\forall$ x,y $\in$ X, if (x,y) $\in$ R and (y,x) $\in$ R then x=y. {Definition 3.3.12}

Problem 14.1 cont.

Symmetric and antisymmetric means no pairwise relation

e.g. xRy doesn't exist if x $\neq$ y

For each element, two ways: self loop or not.

$\Rightarrow$ n$\times$n by Multiplication principle

So, there are n^2 summetric and antisymmetric relations on an n-element set.