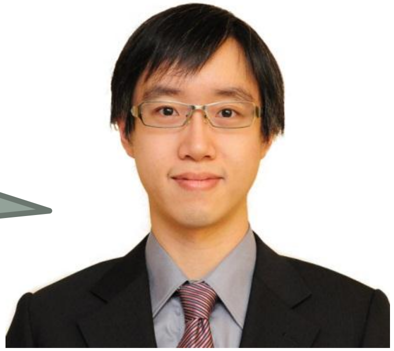
BASICS 2

Prof. Michael Tsai

2012/3/6

Office hour
改成三9-10

作業一的程式題
不容易寫呀!
要趕快開始!



Asymptotic Notation – Omega

- Definition [Omega]:
- $\Omega\big(g(n)\big)$ ={$f(n)$: there exist positive constants c and n_0 such that $0 \leq cg(n) \leq f(n)$ for all $n \geq n_0$}

- $f(n) = \Omega\big(g(n)\big)$
- “f of n is omega of g of n”

- 可以想成是Lower Bound

Examples

- $3n + 2 = \Omega(n)$
- since $3n + 2 \geq 3n$ for all $n \geq 1$.
- $3n + 3 = \Omega(n)$
- since $3n + 3 \geq 3n$ for all $n \geq 1$.
- $100n + 6 = \Omega(n)$
- since $100n + 6 \geq 100n$ for all $n \geq 1$.
- $10n^2 + 4n + 2 = \Omega(n^2)$
- since $10\mathrm{n}^2 + 4\mathrm{n} + 2 \geq n^2$ for all $n \geq 1$.
- $6 * 2^n + n^2 = \Omega(2^n)$
- since $6 * 2^n + n^2 \geq 2^{\mathrm{n}}$ for all $n \geq 1$.

$f(n) \geq cg(n)$ for all $n, n \geq n_0$

Examples

- $3n + 3 = \Omega(1)$
- $10n^2 + 4n + 2 = \Omega(1)$
- $6 * 2^n + n^2 = \Omega(n^{100})$
- $6 * 2^n + n^2 = \Omega(n^{50.2})$
- $6 * 2^n + n^2 = \Omega(n^2)$
- $6 * 2^n + n^2 = \Omega(n)$
- $6 * 2^n + n^2 = \Omega(1)$

$f(n) \geq cg(n)$ for all $n, n \geq n_0$

Discussion

- Omega is a lower bound.
- Should be as large a function as possible.

- Theorem: If $f(n) = a_m n^m + \cdots + a_1 n + a_0$ and $a_m > 0$, then $f(n) = \Omega(n^m)$.

- (證明: 請自己證證看!)

Asymptotic Notation – Theta

- Definition [Theta]:
- $\Theta(g(n)) =$

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

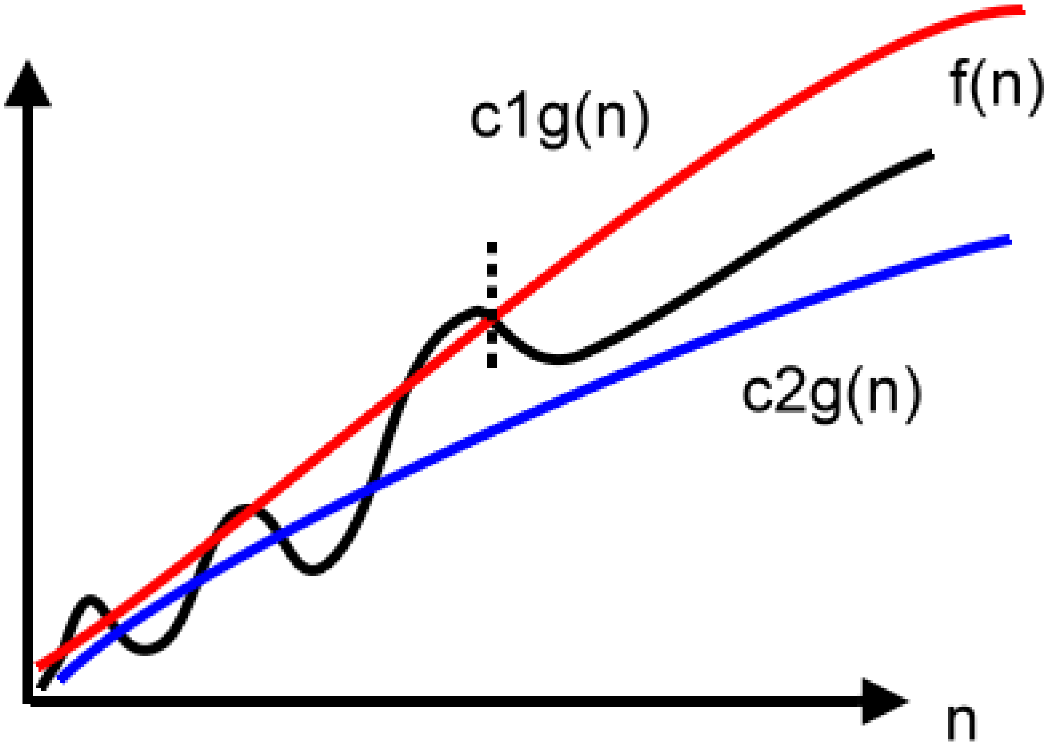
- $f(n) = \Theta(g(n))$
- "f of n is theta of g of n"
- 三明治夾心, 同時具有$O(g(n)$和$\Omega(g(n))$

Theta Platform

GMC Terrain

用圖表示

- Big Oh
- 紅色
- Omega
- 藍色
- Theta
- 紅色藍色都要



Example

- $3n + 2 = \Theta(n)$
- since $3n + 2 \geq 3n$ for all $n \geq 2$ and $3n + 2 \leq 4n$ for all $n \geq 2$.
- $3n + 3 = \Theta(n)$
- $10n^2 + 4n + 2 = \Theta(n^2)$
- $6 * 2^n + n^2 = \Theta(2^n)$
- $10 * \log n + 4 = \Theta(\log n)$
- $3n + 2 \neq \Theta(1)$
- $3n + 3 \neq \Theta(n^2)$
- $10n^2 + 4n + 2 \neq \Theta(n)$
- $10n^2 + 4n + 2 \neq \Theta(1)$
- $6 * 2^n + n^2 \neq \Theta(n^2)$
- $6 * 2^n + n^2 \neq \Theta(n^{100})$
- $6 * 2^n + n^2 \neq \Theta(1)$

$c_1 g(n) \geq f(n) \geq c_2 g(n)$ for all $n, n \geq n_0$

Discussion

- More precise than both the "big oh" and omega notations
- It is true if and only g(n) is both an upper and lower bound on f(n).
- Coefficient = 1
  - $\Omega(6 * 2^n)$ (no good)
  - $\Theta(n^3)$ (ok!)

- Theorem: If $f(n) = a_m n^m + \cdots + a_1 n + a_0$ and $a_m > 0$, then $f(n) = \Theta(n^m)$.
- (證明: 請自己證證看!)

那麼，怎麼知道程式的時間複雜度?

- 看每個statement執行幾次, 畫表來計算
- 有時候複雜度不只跟input大小不一定有關係
- (像是binary search, 跟裡面實際數字是什麼有關)
- 這時候可以取
  - Worst case:給一組input可以使此algorithm執行最慢; 最慢的執行時間是多少?
  - Best case: 給一組input可以使此algorithm執行最快; 最快的執行時間是多少?
  - Average case: 考慮所有case的執行時間, 平均是多少?

- 實際測量程式執行時間最實際!
- (但是常常不可行,此時asymptotic analysis是最好的方法)

Example 1

Statement	Asymptotic complexity
void add(int a[][MAX_SIZE]...) {	0
int i,j;	0
for(i=0;i<rows;i++)	$\Theta(rows)$
for(j-0;j<cols;j++)	$\Theta(rows.cols)$
c[i][j]=a[i][j]+b[i][j];	$\Theta(rows.cols)$
}	0
Total	$\Theta(rows.cols)$

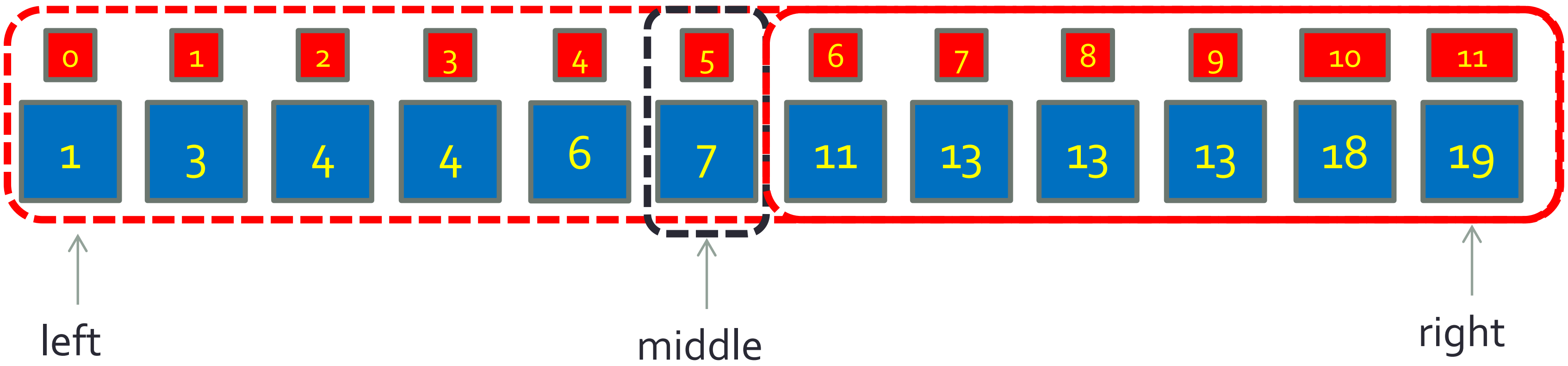
Example 2

```
int binsearch(int list[], int searchnum, int
left, int right) {
   int middle;
   while(left<=right) {
       middle=(left+right)/2;
       switch(COMPARE(list[middle], searchnum)) {
              case -1: left=middle+1;  break;
              case 0: return middle;
              case 1: right=middle-1;
       }
   }
   return -1;
}
```

Worst case: $\Theta(\log n)$

Best case: $\Theta(1)$

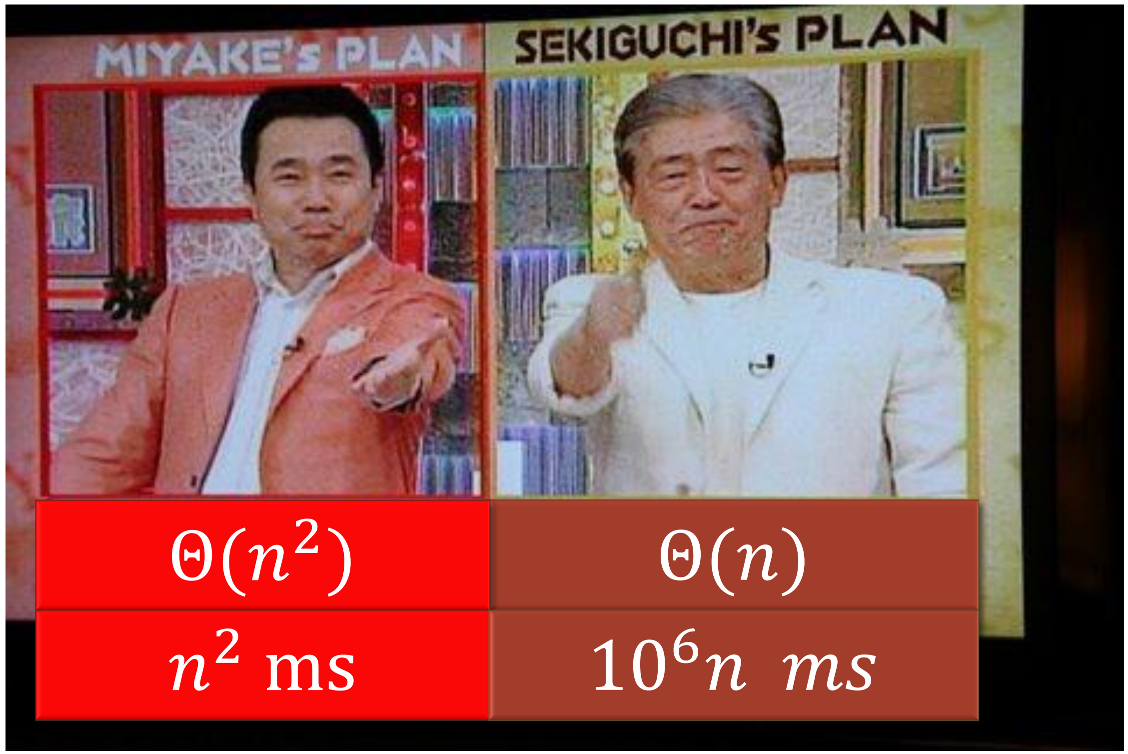
```
searchnum=13;
```




```
middle=(left+right)/2;
left=middle+1;
```

Half the searching window every iteration.

兜基??



$\Theta(n^2)$	$\Theta(n)$
n^2 ms	$10^6 n\ ms$

哪一個好…

$\Theta(n^2)$	$\Theta(n)$
n^2 ms	$10^6 n\ \ ms$

- n很大的時候右邊比左邊好

- 但是n會很大嗎?
- 有時候不會
- 如果n永遠小於10^6 ??
- 結論:要看n大小 (實際上寫程式的時候適用)

Today's Reading Assignments

1. Karumanchi課本1.7-1.18
(特別是1.18, 講解如何分析常見程式結構的running time)

2. 看一下Cormen課本3.1中 small o和 small omega的定定義
(當然整節完整閱讀更好!)