Chapter 2 Basics of Algorithm Analysis

刘骞　51184501130

3.通过观察,我们可以很清楚的知道 f_4 和 f_5 是指数函数，增长速率会比 f_1,f_2,f_3,f_6 增长更快，而在 f_1,f_2,f_3,f_6 中很显然的得出 $f_2=O(f_3)$,且 $f_3=O(f_1)$、$f_3=O(f_6)$,然而多项式函数要比对数函数增长的快，所以 $f_6=O(f_1)$;再看 f_1 显然要小于 f_4 和 f_5，而 f_4 显然有 $f_4=O(f_5)$,所以有 $f_2=O(f_3),f_3=O(f_6),f_6=O(f_1),f_1=O(f_4),f_4=O(f_5)$.

4.通过观察，我们可以分为 g_2、g_6、g_7 和 g_1、g_3、g_4、g_5 两类，显然 $g_3=O(g_4)$,由于在 n 趋于无穷大的时候 $g_1/g_5=0$，所以 $g_1=O(g_5)$,在 n 趋于无穷大的时候 $g_5/g_3=0$，所以 $g_5=O(g_3)$;在 g_2、g_6、g_7 中显然有 $g_7=O(g_6),g_2=O(g_7)$,而在 n 趋于无穷大的时候 $g_4/g_2=0$ 所以 $g_4=O(g_2)$. 所以有 $g_1=O(g_5)$, $g_5=O(g_3$, $g_3=O(g_4)$, $g_4=O(g_2)$, $g_2=O(g_7)$, $g_7=O(g_6)$.

6.

```
for i = 1,    2,…n                                  n
    for j = i+1, i+2,…n                             n
        add up array entries A[i] through A[j]      n
        store the result in B[i, j]
    endfor
endfor
```

(a)　$O(n^3)$

(b)　$\Omega(n^3)$

(c)

i\j	1	2	3	4	5	6	7	….
1	1-2	1-3	1-4	1-5	1-6	1-7	1-8	….
2		2-3	2-4	2-5	2-6	2-7	2-8	….
3			3-4	3-5	3-6	3-7	3-8	….
4				4-5	4-6	4-7	4-8	….
5					5-6	5-7	5-8	….
6						6-7	6-8	….
…							7-8	….

从第二行开始有： $B[i,j]=B[i-1,j]-A[j-1]$

算法伪代码：

/*计算出第一行*/

```
for j = 2,...n                                      n
    add up array entries A[1] through A[j]          n
    store the result in B[1, j]
endfor
```

/*通过第一行的数据规律得到之后每一行的数据*/

```
for i = 2,...n                                      n
    for j = i+1, i+2,...n                           n
        Set B[i-1][j]-A[j-1] to B[i, j]
    endfor
endfor
```

运行时间为： $O(n^2)$