习题课1

苏州大学 计算机科学与技术学院
汪笑宇
Email: xywang21@suda.edu.cn

作业1-1a

2-2 （冒泡排序的正确性） 冒泡排序是一种流行但低效的排序算法，它的作用是反复交换相邻的未按次序排列的元素。

```
BUBBLESORT(A)
for i = 1 to A.length − 1
    for j = A.length downto i + 1
        if A[j] < A[j − 1]
            exchange A[j] with A[j − 1]
```

a. 假设 A'表示 BUBBLESORT(A)的输出。为了证明 BUBBLESORT 正确，我们必须证明它将终止并且有：

$$A'[1] \leqslant A'[2] \leqslant \cdots \leqslant A'[n] \tag{2.3}$$

其中 $n = A.length$。为了证明 BUBBLESORT 确实完成了排序，我们还需要证明什么？下面两部分将证明不等式(2.3)。

- a. 需证明*A*’中的元素与*A*中的元素相同。因为算法中仅对*A*进行了元素交换操作，因此输出的*A*’只是*A*进行元素重排的数组

作业1-1b

2-2 （冒泡排序的正确性） 冒泡排序是一种流行但低效的排序算法，它的作用是反复交换相邻的未按次序排列的元素。

```
BUBBLESORT(A)
1  for i = 1 to A.length − 1
2      for j = A.length downto i + 1
3          if A[j] < A[j − 1]
4              exchange A[j] with A[j − 1]
```

b. 为第 2～4 行的 **for** 循环精确地说明一个循环不变式，并证明该循环不变式成立。你的证明应该使用本章中给出的循环不变式证明的结构。

- b. 循环不变式：$A[j..n]$中最小元素是$A[j]$
  - 初始化：$j=A.length=n$，$A[j..n]$仅包含一个元素$A[j]$，成立
  - 保持：假设$j=k\geq i+1$时成立，即$A[k..n]$最小元素是$A[k]$，那么当$j=k$-1时（此时循环执行时$j=k$，结束后才是$j=k$-1），若$A[k]<A[k$-1$]$则会执行第4行交换操作，此时$A[k$-1$]$为$A[k$-1$..n]$最小元素；反之若$A[k]$大于等于$A[k$-1$]$不执行操作，$A[k$-1$]$依旧为$A[k$-1$..n]$最小元素。因此$j=k$-1 $\geq i$时，$A[k$-1$..n]$中最小元素是$A[k$-1$]$，成立
  - 终止：循环终止时需执行第2行判断，$j=i$即$A[i..n]$中最小元素是$A[i]$

作业1-1c

2-2 （冒泡排序的正确性） 冒泡排序是一种流行但低效的排序算法，它的作用是反复交换相邻的未按次序排列的元素。

```
BUBBLESORT(A)
1  for i = 1 to A.length - 1
2      for j = A.length downto i + 1
3          if A[j] < A[j - 1]
4              exchange A[j] with A[j - 1]
```

c. 使用(b)部分证明的循环不变式的终止条件，为第 1～4 行的 **for** 循环说明一个循环不变式，该不变式将使你能证明不等式(2.3)。你的证明应该使用本章中给出的循环不变式证明的结构。

- c. 循环不变式：$A[1..i]$是A中前i个最小元素且从小到大排序
  - 初始化：i=1，$A[1..i]$仅包含$A[1]$，成立
  - 保持：假设$i=k\leq n-1$时成立，即$A[1]\leq A[2] \leq \ldots \leq A[k]\leq \min\{A[k+1], \ldots, A[n]\}$。则$i=k+1\leq n$时，由b可知：$A[k+1..n]$中最小元素是$A[k+1]$，即$A[k+1]=\min\{A[k+1], \ldots, A[n]\}$。由此可知$A[1..k+1]$是$A$中前$k+1$个最小元素且从小到大排序，成立
  - 终止：循环终止时，$i=A.length=n$，$A[1..n]$是A中前n个最小元素且从小到大排序，即数组A元素从小到大排序

作业1-1d

2-2 （冒泡排序的正确性） 冒泡排序是一种流行但低效的排序算法，它的作用是反复交换相邻的未按次序排列的元素。

```
BUBBLESORT(A)
for i = 1 to A.length − 1
    for j = A.length downto i + 1
        if A[j] < A[j − 1]
            exchange A[j] with A[j − 1]
```

d. 冒泡排序的最坏情况运行时间是多少？与插入排序的运行时间相比，其性能如何？

- d. 代价　执行次数

代价	执行次数
c_1	$T_1 = n = \Theta(n)$
c_2	$T_2 = \sum_{i=1}^{n-1}(n - i + 1) = \Theta(n^2)$
c_3	$T_3 = T_2 - 1 = \Theta(n^2)$
c_4	T_4

- 最坏情况：$T_4 = T_3 = \Theta(n^2)$，总体为$\Theta(n^2)$
- 由于$T_4=O(n^2)$，因此冒泡排序不论最好最坏平均情况运行时间都是$\Theta(n^2)$，而插入排序最好情况运行时间为$\Theta(n)$

作业1-2(1) (a)-(c)

$f(n)$和$g(n)$为渐近正函数，证明：
(a) $O(f(n))+O(g(n))=O(\max\{f(n), g(n)\})$
(b) $O(f(n))+O(g(n))=O(f(n)+g(n))$
(c) $O(f(n))\cdot O(g(n))=O(f(n)\cdot g(n))$

- 证明：设$h_1(n)=O(f(n))$, $h_2(n)=O(g(n))$，且h_1, h_2均为渐近正函数，则根据定义可找到一个$n_0>0$以及相应的常数c_1, $c_2>0$，使得当$n\geq n_0$时同时满足$0\leq h_1(n)\leq c_1 f(n)$以及$0\leq h_2(n)\leq c_2 g(n)$
  - (a) 当$n\geq n_0$时，$0\leq h_1(n)+h_2(n)\leq c_1 f(n)+c_2 g(n)\leq c_1\max\{f(n), g(n)\}+c_2\max\{f(n), g(n)\}=(c_1+c_2)\max\{f(n), g(n)\}$。因此取$c_3=c_1+c_2$即得当$n\geq n_0$时，$0\leq h_1(n)+h_2(n)\leq c_3\max\{f(n), g(n)\}$，即$O(f(n))+O(g(n))=O(\max\{f(n), g(n)\})$
  - (b) 当$n\geq n_0$时，$0\leq h_1(n)+h_2(n)\leq c_1 f(n)+c_2 g(n)\leq\max\{c_1, c_2\}f(n)+\max\{c_1, c_2\}g(n)=\max\{c_1, c_2\}(f(n)+g(n))$。因此取$c_3=\max\{c_1, c_2\}$即得当$n\geq n_0$时，$0\leq h_1(n)+h_2(n)\leq c_3(f(n)+g(n))$，即$O(f(n))+O(g(n))=O(f(n)+g(n))$
  - (c) 当$n\geq n_0$时，$0\leq h_1(n)\cdot h_2(n)\leq c_1 f(n)\cdot c_2 g(n)\leq c_1 c_2 f(n)\cdot g(n)$。因此取$c_3=c_1 c_2$即得当$n\geq n_0$时，$0\leq h_1(n)\cdot h_2(n)\leq c_3(f(n)\cdot g(n))$，即$O(f(n))\cdot O(g(n))=O(f(n)\cdot g(n))$

作业1-2(1) (d)-(e)

$f(n)$和$g(n)$为渐近正函数，$c>0$是一个常量，证明：
(d) $O(cf(n))=O(f(n))$
(e) $g(n)=O(f(n))$蕴涵$O(f(n))+O(g(n))=O(f(n))$

- 证明：设$h_1(n)=O(f(n)), h_2(n)=O(g(n))$，且$h_1, h_2$均为渐近正函数，则根据定义可找到一个$n_0>0$以及相应的常数$c_1, c_2>0$，使得当$n\geq n_0$时同时满足$0\leq h_1(n)\leq c_1f(n)$以及$0\leq h_2(n)\leq c_2g(n)$
  - (d) 设$h_3(n)=O(cf(n))$为渐近正函数，则根据定义可找到一个$n_1>0$以及相应的常数$c_4>0$，使得当$n\geq n_1$时满足$0\leq h_3(n)\leq c_4cf(n)$。因此取$c_3=c_4c$即得当$n\geq n_1$时，$0\leq h_3(n)\leq c_3f(n)$，即$O(cf(n))=O(f(n))$
  - (e) 因为$g(n)=O(f(n))$，所以可找到一个$n_2>0$以及相应的常数$c_5>0$，使得当$n\geq n_2$时满足$0\leq g(n)\leq c_5f(n)$。取$n_3=\max\{n_0, n_2\}$，可得当$n\geq n_3$时，有$0\leq h_1(n)+h_2(n)\leq c_1f(n)+c_2g(n)\leq c_1f(n)+c_2c_5f(n)=(c_1+c_2c_5)\, f(n)$。因此取$c_3=c_1+c_2c_5$即得当$n\geq n_3$时，$0\leq h_1(n)+h_2(n)\leq c_3\, f(n)$，即$O(f(n))+O(g(n))=O(f(n))$

作业1-2(2)

3-4 （渐近记号的性质） 假设 $f(n)$和 $g(n)$为渐近正函数。证明或反驳下面的每个猜测。

a. $f(n)=O(g(n))$蕴涵 $g(n)=O(f(n))$。

b. $f(n)+g(n)=\Theta(\min(f(n),\ g(n)))$。

c. $f(n)=O(g(n))$蕴涵 $\lg(f(n))=O(\lg(g(n)))$，其中对所有足够大的 n，有 $\lg(g(n))\geqslant 1$ 且 $f(n)\geqslant 1$。

d. $f(n)=O(g(n))$蕴涵 $2^{f(n)}=O(2^{g(n)})$。

e. $f(n)=O((f(n))^2)$。

f. $f(n)=O(g(n))$蕴涵 $g(n)=\Omega(f(n))$。

g. $f(n)=\Theta(f(n/2))$。

h. $f(n)+o(f(n))=\Theta(f(n))$。

- c, f, h正确，其他全部错误
  - 反驳：
    - a, b: $f(n)=n,\ g(n)=n^2$
    - d: $f(n)=2n,\ g(n)=n$，$2^{2n}=4^n\neq O(2^n)$
    - e: $f(n)=1/n$
    - g: $f(n)=4^n$，$f(n/2)=2^n$，$4^n\neq\Theta(2^n)$

作业1-2(2)c

3-4 （渐近记号的性质） 假设 $f(n)$和 $g(n)$为渐近正函数。证明或反驳下面的每个猜测。

a. $f(n)=O(g(n))$蕴涵 $g(n)=O(f(n))$。

b. $f(n)+g(n)=\Theta(\min(f(n),\ g(n)))$。

c. $f(n)=O(g(n))$蕴涵 $\lg(f(n))=O(\lg(g(n)))$，其中对所有足够大的 n，有 $\lg(g(n))\geqslant 1$ 且 $f(n)\geqslant 1$。

d. $f(n)=O(g(n))$蕴涵 $2^{f(n)}=O(2^{g(n)})$。

e. $f(n)=O((f(n))^2)$。

f. $f(n)=O(g(n))$蕴涵 $g(n)=\Omega(f(n))$。

g. $f(n)=\Theta(f(n/2))$。

h. $f(n)+o(f(n))=\Theta(f(n))$。

- c. 证明：根据定义可找到一个足够大的$n_0>0$以及相应的常数$c>0$，使得当$n\geq n_0$时满足$0\leq f(n)\leq cg(n)$，且$\lg(g(n))\geq 1$且$f(n)\geq 1$，因此当$n\geq n_0$时满足$0\leq \lg(f(n))\leq \lg(cg(n))=\lg c+\lg(g(n))\leq(\lg c+1)\lg(g(n))$。因此取$c_1=\lg c+1$，即得当$n\geq n_0$时$0\leq \lg(f(n))\leq c_1\lg(g(n))$，即$\lg(f(n))=O(\lg(g(n)))$

作业1-2(2)f

3-4 （渐近记号的性质） 假设 $f(n)$ 和 $g(n)$ 为渐近正函数。证明或反驳下面的每个猜测。

a. $f(n)=O(g(n))$ 蕴涵 $g(n)=O(f(n))$。

b. $f(n)+g(n)=\Theta(\min(f(n), g(n)))$。

c. $f(n)=O(g(n))$ 蕴涵 $\lg(f(n))=O(\lg(g(n)))$，其中对所有足够大的 n，有 $\lg(g(n))\geqslant 1$ 且 $f(n)\geqslant 1$。

d. $f(n)=O(g(n))$ 蕴涵 $2^{f(n)}=O(2^{g(n)})$。

e. $f(n)=O((f(n))^2)$。

f. $f(n)=O(g(n))$ 蕴涵 $g(n)=\Omega(f(n))$。

g. $f(n)=\Theta(f(n/2))$。

h. $f(n)+o(f(n))=\Theta(f(n))$。

- f. 证明：根据定义可找到 $n_0>0$ 以及相应的常数 $c>0$，使得当 $n\geq n_0$ 时满足 $0\leq f(n)\leq cg(n)$，即 $0\leq(1/c)f(n)\leq g(n)$。因此取 $c_1=1/c$，即得当 $n\geq n_0$ 时，$0\leq c_1 f(n)\leq g(n)$，即 $g(n)=\Omega(f(n))$

作业1-2(2)h

3-4 （渐近记号的性质） 假设 $f(n)$和 $g(n)$为渐近正函数。证明或反驳下面的每个猜测。

a. $f(n)=O(g(n))$蕴涵 $g(n)=O(f(n))$。

b. $f(n)+g(n)=\Theta(\min(f(n),\ g(n)))$。

c. $f(n)=O(g(n))$蕴涵 $\lg(f(n))=O(\lg(g(n)))$，其中对所有足够大的 n，有 $\lg(g(n))\geqslant 1$ 且 $f(n)\geqslant 1$。

d. $f(n)=O(g(n))$蕴涵 $2^{f(n)}=O(2^{g(n)})$。

e. $f(n)=O((f(n))^2)$。

f. $f(n)=O(g(n))$蕴涵 $g(n)=\Omega(f(n))$。

g. $f(n)=\Theta(f(n/2))$。

h. $f(n)+o(f(n))=\Theta(f(n))$。

- h. 证明：设$h(n)=o(f(n))$为渐近正函数，由定义可知对任意常数$c>0$，存在$n_0>0$，使得当$n\geq n_0$时有$0\leq h(n)<cf(n)$。因此当$n\geq n_0$时， $0\leq f(n)\leq f(n)+h(n)<f(n)+cf(n)=(c+1)f(n)$，其中最后一个小于号可写为小于等于。由于$c$可取任意大于0的常数，因此可取$c_1=1, c_2=2$，当$n\geq n_0$时有 $0\leq c_1f(n)\leq f(n)+o(f(n))\leq c_2f(n)$，即$f(n)+o(f(n))=\Theta(f(n))$

作业1-3

■排序结果：$\{3, 10^{10}\}$, $\lg\lg n$, $5n$, $\{\lg(n!), n\lg(n+2)\}$, n^3, n^{100}, $n^{\lg n}$, $(3/2)^n$, 2^n, 2^{2n}, $n7^n$, $n!$, n^n, 2^{2^n}

➢采用相除求极限的方法证明排序，可使用洛必达法则

➢例：

$$\lim_{n\to\infty}\frac{5n}{\lg\lg n}=\lim_{n\to\infty}\frac{5}{\frac{1}{\lg n\cdot\ln 2}\cdot\frac{1}{n\ln 2}}=\lim_{n\to\infty}(\ln 2)^2 n\lg n=\infty$$

$$\lim_{n\to\infty}\frac{2^{2n}}{2^n}=\lim_{n\to\infty}\frac{4^n}{2^n}=\lim_{n\to\infty}2^n=\infty$$

强调：本课程中约定lg函数表示以2为底的对数！！！（参见教材第32页）

作业1-3 (续)

- 排序结果：$\{3, 10^{10}\}$, $\lg\lg n$, $5n$, $\{\lg(n!), n\lg(n+2)\}$, n^3, n^{100}, $n^{\lg n}$, $(3/2)^n$, 2^n, 2^{2n}, $n7^n$, $n!$, n^n, 2^{2^n}
  - 以下说明$n^{\lg n}$大于任意多项式阶，且小于任何大于1为底数的指数阶，即$n^{\lg n}=\omega(n^d)$且$n^{\lg n}=o(k^n)$其中$k>1$

$$\lim_{n\to\infty}\frac{n^d}{n^{\lg n}}=\lim_{n\to\infty}n^{d-\lg n}=0$$

$$\lim_{n\to\infty}\frac{k^n}{n^{\lg n}}=\lim_{n\to\infty}\frac{k^n}{k^{\log_k n\lg n}}=\lim_{n\to\infty}k^{\boxed{n-\log_k n\lg n}}=\infty$$

设函数$f(n)=n-\log_k n\lg n$，则$f'(n)=1-\left(\boxed{\frac{\lg n}{n\ln k}}+\boxed{\frac{\log_k n}{n\ln 2}}\right)$

由于$\frac{\log_k n}{n}$在n趋于无穷大时趋向于0

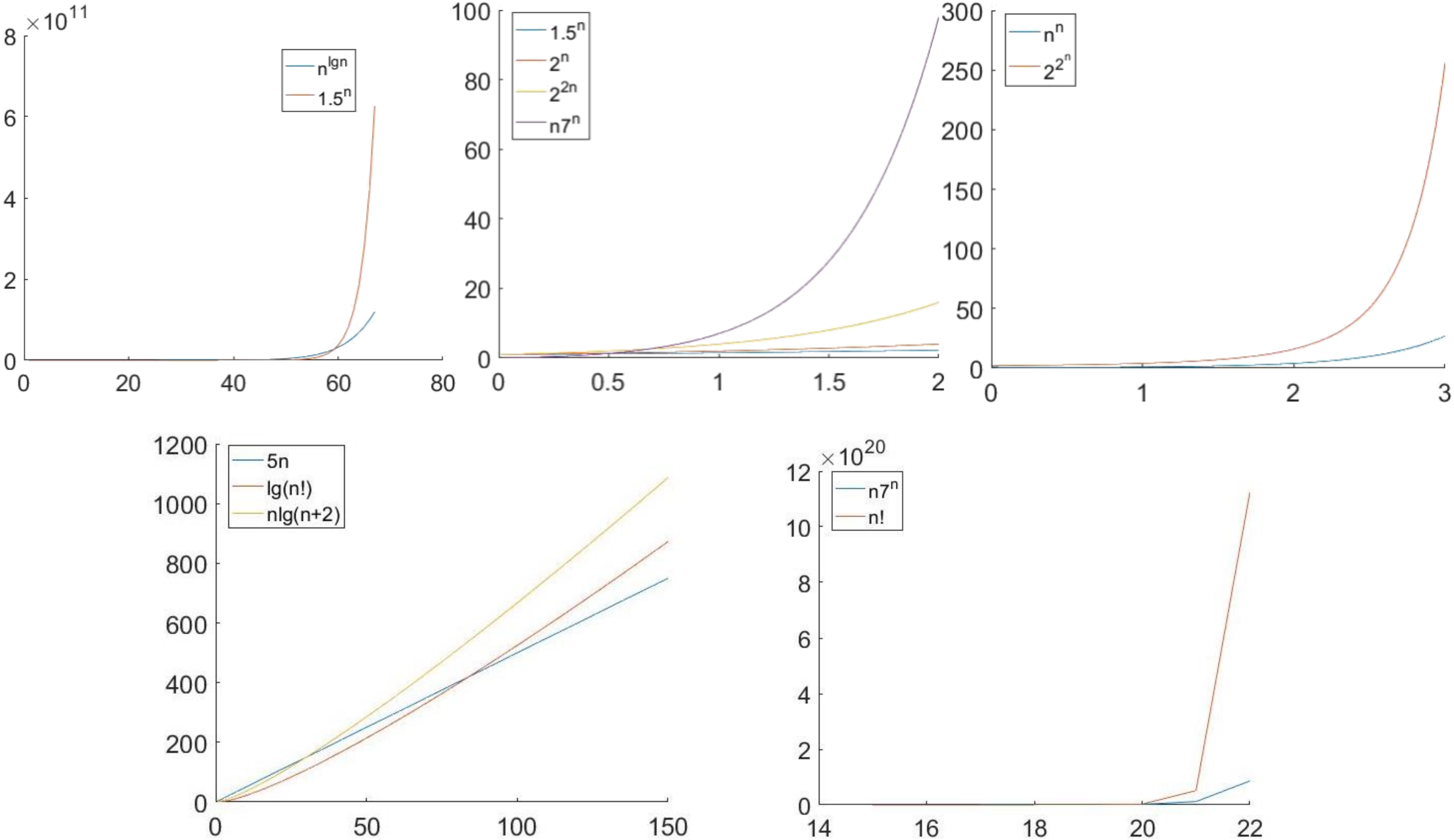
所以在n趋于无穷大时$f(n)$单调递增，且递增斜率趋近于1 $\Rightarrow \lim_{n\to\infty}f(n)=\infty$

作业1-3 (续)

- 排序结果：$\{3, 10^{10}\}$, $\lg\lg n$, $5n$, $\{\lg(n!), n\lg(n+2)\}$, n^3, n^{100}, $n^{\lg n}$, $(3/2)^n$, 2^n, 2^{2n}, $n7^n$, $n!$, n^n, 2^{2^n}
  - 以下说明n^n和2^{2^n}的渐近增长关系

$$\lim_{n\to\infty}\frac{2^{2^n}}{n^n}=\lim_{n\to\infty}\frac{2^{2^n}}{2^{n\lg n}}=\lim_{n\to\infty}2^{2^n-n\lg n}=\infty$$

作业1-3 (续)


$\times 10^{11}$
n^{lgn}
1.5^n
1.5^n
2^n
2^{2n}
$n7^n$
n^n
2^{2^n}
5n
lg(n!)
nlg(n+2)
$\times 10^{20}$
$n7^n$
n!

作业1-4

- 证明以下性质成立：$a^{\log_b c} = c^{\log_b a}$，
  其中$a, b, c>0$且$b \neq 1$
- $a^{\log_b c} = b^{\log_b (a^{\log_b c})} = b^{\log_b c \log_b a} = b^{\log_b (c^{\log_b a})} = c^{\log_b a}$

作业1-5(a)

- 证明 (a) lg(n!)=Θ($n$lg$n$)
- 使用斯特林公式证明
  - 证明：(a)

$$
\begin{aligned}
\lg(n!) &= \lg(\sqrt{2\pi n}(n/e)^n(1+\Theta(1/n))) \\
&= \lg(2\pi n)/2 + n\lg(n/e) + \lg(1+\Theta(1/n)) \\
&= \lg(2\pi)/2 + (\lg n)/2 + n\lg n - n\lg e + \lg(1+\Theta(1/n)) \\
&= \Theta(1) + \Theta(\lg n) + \Theta(n\lg n) - \Theta(n) \\
&= \Theta(n\lg n)
\end{aligned}
$$

作业1-5(b)

■证明 (b) $n!=\omega(2^n)$且$n!=o(n^n)$

➢证明：(b) $$\lim_{n\to\infty}\frac{n!}{2^n}=\lim_{n\to\infty}\frac{\sqrt{2\pi n}(n/e)^n(1+\Theta(1/n))}{2^n}$$
$$=\lim_{n\to\infty}\sqrt{2\pi n}(1+\Theta(1/n))\left(\frac{n}{2e}\right)^n$$
$$=\lim_{n\to\infty}\sqrt{2\pi n}\left(\frac{n}{2e}\right)^n$$
$$=\infty \qquad (n>2e)$$

可推导出对所有的k>0，都有$n!=\omega(k^n)$

$$\lim_{n\to\infty}\frac{n!}{n^n}=\lim_{n\to\infty}\frac{\sqrt{2\pi n}(n/e)^n(1+\Theta(1/n))}{n^n}$$
$$=\lim_{n\to\infty}\sqrt{2\pi}(1+\Theta(1/n))\left(\frac{\sqrt{n}}{e^n}\right)$$
$$=\lim_{n\to\infty}\sqrt{2\pi}\cdot\left(\frac{\sqrt{n}}{e^n}\right)$$
$$=0$$

作业2-1

■4.4-2 $T(n)=T(n/2)+n^2$

树高 $\lg n$		代价
	n^2	n^2
	$n^2/4$	$n^2/4$
	$n^2/16$	$n^2/16$
	$\vdots$	$\vdots$
		$n^2/(2^{\lg n-1})^2=4$
	$\Theta(1)$	$\Theta(1)$

$$
\begin{aligned}
T(n) &= n^2 + \frac{n^2}{4} + \frac{n^2}{4^2} + \ldots + \frac{n^2}{4^{\lg n - 1}} + \Theta(1) \\
&= n^2 \cdot \frac{1-(1/4)^{\lg n}}{1-1/4} + \Theta(1) \\
&< \frac{4}{3}n^2 + \Theta(1) = O(n^2)
\end{aligned}
$$

代入法验证：

即需证明可恰当选择常数$c>0$，使得$T(n)\leq cn^2$

假设对于所有的$m<n$都有$T(m)\leq cm^2$，特别的，对于$m=n/2$有$T(n/2)\leq cn^2/4$，则

$T(n)\leq cn^2/4+n^2=(c/4+1)n^2\leq cn^2$

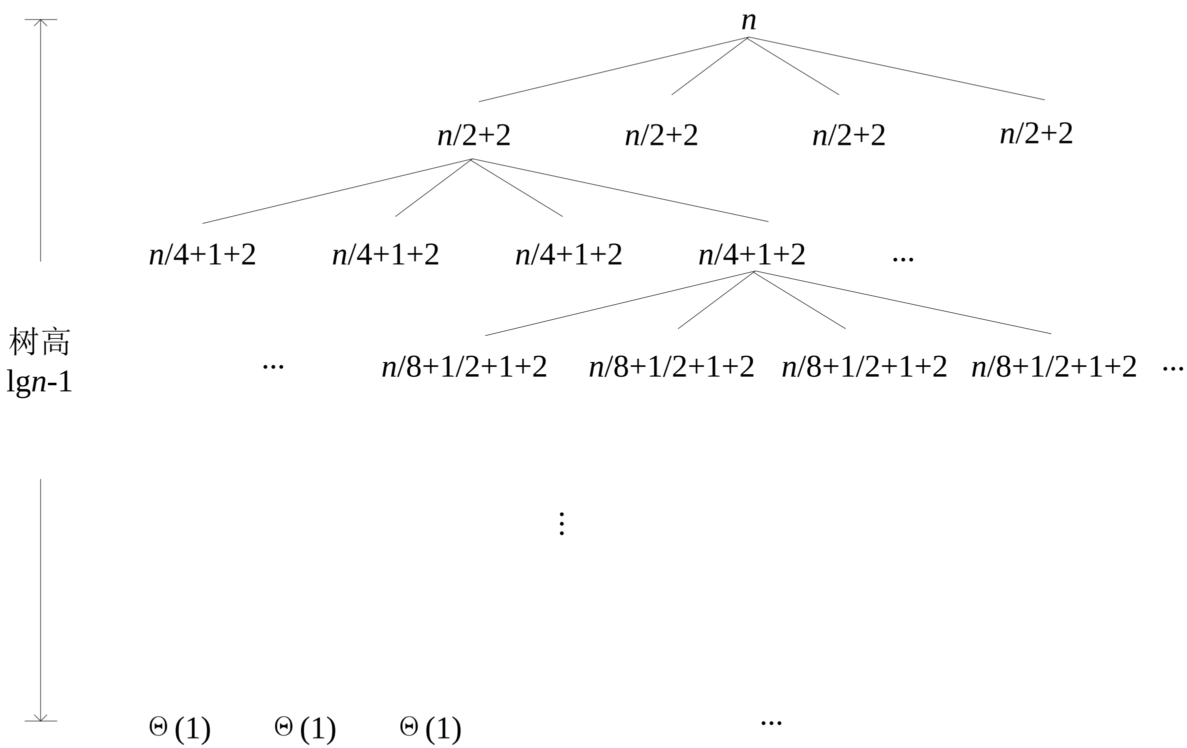
当$c\geq 4/3$时成立

取$c=4/3$，边界条件$T(1)=1\leq 4/3$满足条件

因此$T(n)=O(n^2)$成立

作业2-1 (续)

■4.4-3 $T(n)=4T(n/2+2)+n$



代价

n

$4(n/2+2)$

$4^2(n/4+1+2)$

$4^3(n/8+1/2+1+2)$

$\vdots$

$$4^{\lg n-2}(n/2^{\lg n-2}+\sum_{i=0}^{\lg n-3}2\times(1/2)^i)$$

$$=4^{\lg n-2}(n/2^{\lg n-2}+\frac{2-2\times(1/2)^{\lg n-2}}{1-1/2})$$

$$=4^{\lg n-2}(n/2^{\lg n-2}+4-4/2^{\lg n-2})$$

$\Theta(4^{\lg n-1})=\Theta(n^2/4)=\Theta(n^2)$

n = 1,2,3,4时都是递归边界条件，否则递归无法终止或导致矛盾

假设：$T(1)=T(2)=T(3)=T(4)=1$

作业2-1 (续)

■4.4-3 $T(n)=4T(n/2+2)+n$

$$
\begin{aligned}
T(n) &= n + 4(n/2+2) + \ldots + 4^{\lg n-2}(n/2^{\lg n-2} + 4 - 4/2^{\lg n-2}) + \Theta(n^2) \\
&= \sum_{i=0}^{\lg n-2} 4^i((n-4)/2^i + 4) + \Theta(n^2) \\
&= \sum_{i=0}^{\lg n-2} 2^i n - \sum_{i=0}^{\lg n-2} 2^{i+2} + \sum_{i=0}^{\lg n-2} 4^{i+1} + \Theta(n^2) \\
&= n \cdot \frac{1-2^{\lg n-1}}{1-2} - \frac{4-2^{\lg n+1}}{1-2} + \frac{4-4^{\lg n}}{1-4} + \Theta(n^2) \\
&= n(n/2-1) - (2n-4) + (n^2-4)/3 + \Theta(n^2) \\
&= 5n^2/6 - 3n + \Theta(n^2) \\
&= O(n^2)
\end{aligned}
$$

作业2-1 (续)

■4.4-3 $T(n)=4T(n/2+2)+n$

代入法验证：（加入修正项）
证明可恰当选择常数$c,d>0$，使得$T(n)\le c(n-4)^2-d(n-4)$
假设对于所有的$m<n$都有$T(m)\le c(m-4)^2-d(m-4)$，特别的对于$m=n/2+2<n$，有
$T(n/2+2) \le c(n/2+2-4)^2-d(n/2+2-4) = c(n/2-2)^2-d(n/2-2)$，其中$n\ge 5$
则
$T(n) \le 4(c(n/2-2)^2-d(n/2-2))+n = c((2(n/2-2))^2)-2d(2(n/2-2))+n= c(n-4)^2-2d(n-4)+n$
$\le c(n-4)^2-d(n-4)$
当c为任意正数且$-2d(n-4)+n\le -d(n-4)$即$d\ge 1/(1-4/n)$时成立。因$n\ge 5$，则$d\ge 5$。
因$T(5)= 4T(4)+5=9 \le c-d$，
取$c=14$, $d=5$, $n_0=5$，边界条件$T(5) = 9 \le 14*(5-4)^2-5*(5-4)=9$满足条件
因此$T(n) = O(n^2)$成立

作业2-1 (续)

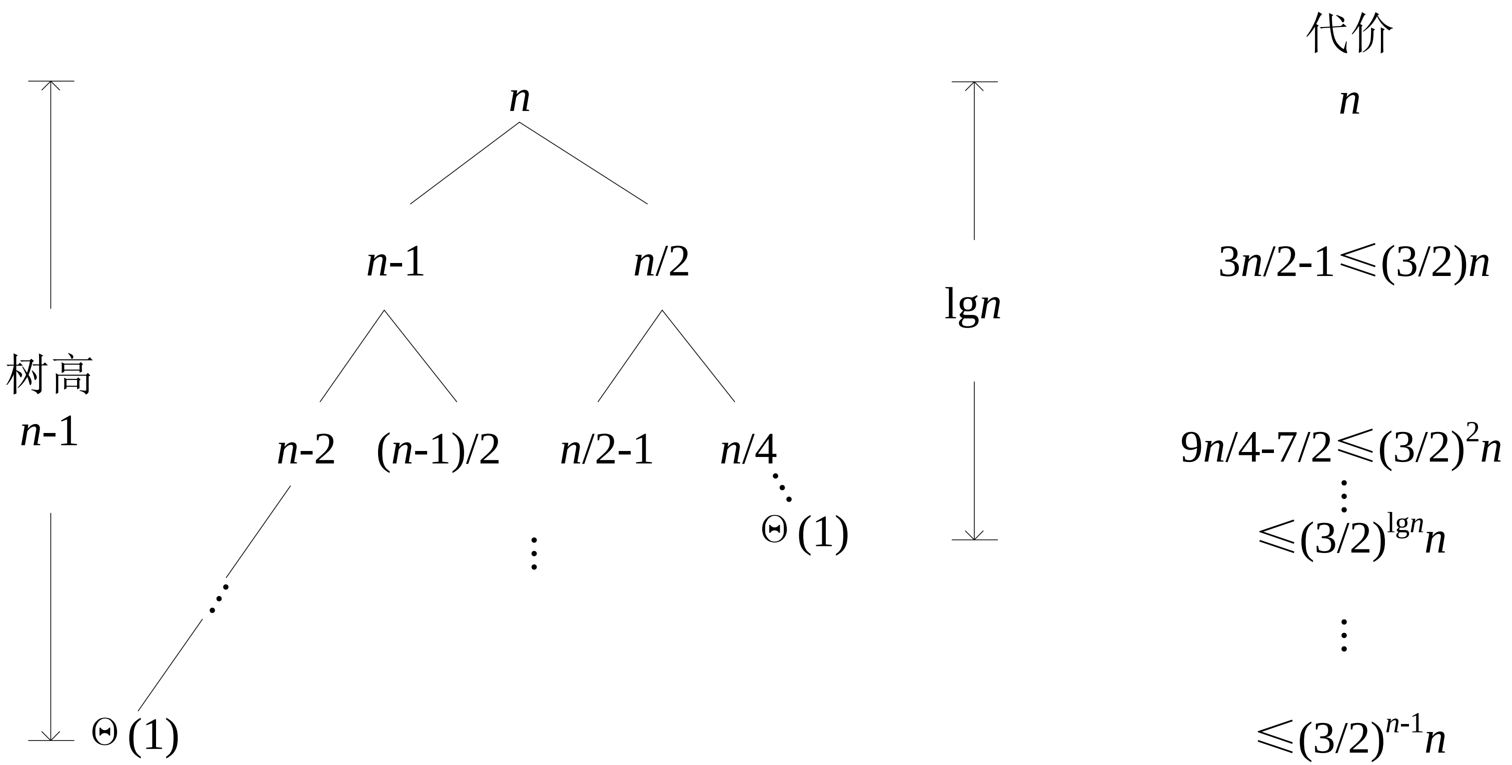
■4.4-4 $T(n)=T(n-1)+1$

$$T(n) = T(n-1) + 1 = T(n-2) + 2 = T(n-3) + 3$$
$$= \ldots = T(n-(n-1)) + (n-1) = O(n)$$

代入法验证：
即需证明可恰当选择常数$c>0$，使得$T(n)\le cn$
假设对于所有的$m<n$都有$T(m)\le cm$，
特别的，对于$m=n-1$ 有$T(n-1) \le c(n-1)$，则
$T(n) \le c(n-1)+1 = cn-(c-1) \le cn$
当$c\ge 1$时成立
取$c=1$，边界条件$T(1) = 1 \le 1$满足条件
因此$T(n) = O(n)$成立

作业2-1 (续)

■4.4-5 $T(n)=T(n-1)+T(n/2)+n$



作业2-1 (续)

■4.4-5 $T(n)=T(n-1)+T(n/2)+n$

$$T(n) < \sum_{i=0}^{n-1}(3/2)^i n$$

$$= n \cdot \frac{1-(3/2)^n}{1-3/2} = 2n\cdot(3/2)^n - 2n = O(2^n)$$

代入法验证：

即需证明可恰当选择常数$c>0$，使得$T(n)\le c2^n$

假设对于所有的$m<n$都有$T(m)\le c2^m$，则

$T(n) \le c2^{n-1}+c2^{n/2}+n = c/2\cdot 2^n + c(\sqrt{2})^n + n \le c2^n$

当$n\ge 3$且$c\ge \frac{3}{4-2\sqrt{2}} \approx 2.56$ 时成立

取$c=3$，边界条件$T(1) = 1 \le 6$, $T(2) = 2T(1)+2 = 4 \le 12$

满足条件

因此$T(n) = O(2^n)$成立

作业2-2 a-c

4-1 （递归式例子） 对下列每个递归式，给出 $T(n)$的渐近上界和下界。假定 $n\leqslant 2$ 时 $T(n)$是常数。给出尽量紧确的界，并验证其正确性。

a. $T(n)=2T(n/2)+n^4$

b. $T(n)=T(7n/10)+n$

c. $T(n)=16T(n/4)+n^2$

d. $T(n)=7T(n/3)+n^2$

e. $T(n)=7T(n/2)+n^2$

f. $T(n)=2T(n/4)+\sqrt{n}$

依照递归式形式$T(n) = aT(n/b) + f(n)$进行说明：

a. $a=2, b=2, n^{\log_b a}=n^{\lg 2}=n, f(n)=n^4=\Omega(n^4)=\Omega(n^{1+\epsilon})$

其中ε=4–1=3>0。当c=1/8且n足够大时，$af(n/b) = 2(n/2)^4=n^4/8 \leq n^4/8 = cf(n)$

根据主定理第三种情况，$T(n) = \Theta(n^4)$

b. $a=1, b=10/7, n^{\log_b a}=n^{\log_{\frac{10}{7}} 1}=1, f(n)=n=\Omega(n)=\Omega(n^{0+\epsilon})$

其中ε=1–0=1>0。当c=7/10且n足够大时，$af(n/b) = 7n/10 \leq 7n/10 = cf(n)$

根据主定理第三种情况，$T(n) = \Theta(n)$

c. $a=16, b=4, n^{\log_b a}=n^{\log_4 16}=n^2, f(n)=n^2$，根据主定理第二种情况，$T(n) = \Theta(n^2\lg n)$

作业2-2 d-f

4-1 （递归式例子） 对下列每个递归式，给出 $T(n)$ 的渐近上界和下界。假定 $n \leqslant 2$ 时 $T(n)$ 是常数。给出尽量紧确的界，并验证其正确性。

a. $T(n)=2T(n/2)+n^4$

b. $T(n)=T(7n/10)+n$

c. $T(n)=16T(n/4)+n^2$

d. $T(n)=7T(n/3)+n^2$

e. $T(n)=7T(n/2)+n^2$

f. $T(n)=2T(n/4)+\sqrt{n}$

依照递归式形式$T(n) = aT(n/b) + f(n)$进行说明：

d. $a = 7, b = 3, n^{\log_b a} = n^{\log_3 7}, f(n) = n^2 = \Omega(n^2) = \Omega(n^{\log_3 7+\epsilon})$

其中ε=2–$\log_3 7$>0。当c=7/9且n足够大时，$af(n/b) = 7(n/3)^2=7n^2/9 \leq 7n^2/9 = cf(n)$

根据主定理第三种情况，$T(n) = \Theta(n^2)$

e. $a = 7, b = 2, n^{\log_b a} = n^{\lg 7}, f(n) = n^2 = O(n^{\lg 7}) = O(n^{2+\epsilon})$

其中ε=lg7–2>0。根据主定理第一种情况，$T(n) = \Theta(n^{\lg 7})$

f. $a = 2, b = 4, n^{\log_b a} = n^{\log_4 2} = n^{1/2}, f(n) = n^{1/2}$，根据主定理第二种情况，

$T(n) = \Theta(\sqrt{n}\lg n)$

作业2-3

- 受限汉诺塔问题：有3个塔座从左到右分别为X、Y、Z，每次只能将一个塔座上最小的圆盘移动到相邻塔座上，即不可将圆盘从X直接移动到Z（或者从Z 直接移动到X）。当然，依旧不允许大圆盘放在小圆盘上。任务为：将n个大小不同且依次叠放的圆盘从X 移动到Z。
- 请设计一个递归算法：(1) 说明算法设计思想，(2) 写出伪代码，(3) 分析使用该算法需移动圆盘的次数。

作业2-3 (续)

- 设计思想：每次只能移动到相邻塔座，因此最下方圆盘需要先从X移动到Y，再从Y移动到Z。由此，上方$n-1$个圆盘要先移动到Z，待最下方圆盘移走后再移回X，最下方圆盘移动到Z后，上方圆盘再移动到Z。

```
RESTRICTED_HANOI(n, X, Y, Z)
if n = 0
    return
else RESTRICTED_HANOI(n – 1, X, Y, Z)  //上方n – 1个圆盘移动到Z
    move(n, X, Y)  //n号圆盘移动到Y
    RESTRICTED_HANOI(n – 1, Z, Y, X)   //上方n – 1个圆盘移动到X
    move(n, Y, Z)   //n号圆盘移动到Z
    RESTRICTED_HANOI(n – 1, X, Y, Z)   //上方n – 1个圆盘移动到Z
```

作业2-3 (续)

■移动次数：$T(n) = 3T(n-1) + 2$，其中$T(0) = 0$

$$
\begin{aligned}
T(n) &= 3T(n-1) + 2 \\
&= 3(3T(n-2) + 2) + 2 \\
&= 3(3(3T(n-3) + 2) + 2) + 2 \\
&\ \ \vdots \\
&= 3^n T(n-n) + 2 \cdot (3^0 + 3^1 + \ldots + 3^{n-1}) \\
&= 3^n - 1 \\
&= O(3^n)
\end{aligned}
$$

作业2-4

2-4 （逆序对） 假设$A[1..n]$是一个有n个不同数的数组。若$i<j$且$A[i]>A[j]$，则对偶(i, j)称为A的一个**逆序对**(inversion)。

a. 列出数组〈2，3，8，6，1〉的5个逆序对。

b. 由集合$\{1, 2, \cdots, n\}$中的元素构成的什么数组具有最多的逆序对？它有多少逆序对？

c. 插入排序的运行时间与输入数组中逆序对的数量之间是什么关系？证明你的回答。

d. 给出一个确定在n个元素的任何排列中逆序对数量的算法，最坏情况需要$\Theta(n \lg n)$时间。（提示：修改归并排序。）

a. (1, 5), (2, 5), (3, 5), (4, 5), (3, 4)

b. 从大到小排序的数组<n, n-1, n-2, ..., 2, 1>具有最多逆序对，任两个元素都构成逆序对，因此总共有$C_n^2 = n(n-1)/2$个逆序对

c. 插入排序运行时间与输入数组中逆序对数量呈线性关系

作业2-4 (续)

	代价	次数
INSERTION_SORT(A)		
1 **for** $j \leftarrow 2$ **to** $A.length$ **do**	c_1	n
2 $\quad key \leftarrow A[j]$	c_2	n-1
3 $\quad$// Insert $A[j]$ into the sorted sequence $A[1..j$-1]	0	n-1
4 $\quad i \leftarrow j - 1$	c_4	n-1
5 $\quad$ **while** i>0 and $\boxed{A[i]>key}$ **do**	c_5	$\sum_{j=2}^{n} t_j$
6 $\qquad A[i+1] \leftarrow A[i]$	c_6	$\sum_{j=2}^{n}(t_j - 1)$
7 $\qquad i \leftarrow i - 1$	c_7	$\sum_{j=2}^{n}(t_j - 1)$
8 $\quad A[i+1] \leftarrow key$	c_8	n-1

t_j：对于值j，第5行执行while循环测试的次数

$A[i]>key$满足一次条件则存在一个逆序对(i, j)，因此while循环的循环体执行一次则存在一个逆序对，即$I(n) = \sum_{j=2}^{n}(t_j - 1)$

由此可写出总代价$T(n)=aI(n)+f(n)$，其中a是一个常数，且当n确定时$f(n)$也是一个常数，因此与逆序对数量关系呈线性关系

作业2-4 (续)

d. 基本思想：归并排序在合并阶段可对逆序对进行统计，若右半子数组元素$A[j]$较左半子数组元素$A[i]$先加入最终序列，则说明当前元素$A[j]$与左半子数组剩下的所有元素都构成逆序对，此时逆序对数量加上左半子数组剩下元素数量$n_1 - i + 1$

```
MERGE_INVERSION(A, p, q, r)
n1 ← q – p + 1
n2 ← r – q
let L[1..n1+1] and R[1..n2+1] be new arrays
for i ← 1 to n1 do L[i] ← A[p+i–1]
for j ← 1 to n2 do R[j] ← A[q+j]
L[n1+1] ← ∞,  R[n2+1] ← ∞
i ← 1,  j ← 1, inv ← 0
for k ← p to r do
    if L[i] ≤ R[j]
        A[k] ← L[i],  i ← i + 1
    else A[k] ← R[j],  j ← j + 1, inv ← inv + n1 – i + 1
return inv
```

```
INVERSION(A, p, r)
if p < r
    q ← ⌊(p + r)/2⌋
    left ← INVERSION(A, p, q)
    right ← INVERSION(A, q+1, r)
    return left + right + MERGE_ INVERSION(A, p, q, r)
```