Homework 1

和泳毅 PB19010450

1.假定$f(n)$与$g(n)$都是渐进非负函数，判断下列等式或陈述是否一定是正确的，并简要解释你的答案.

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

错误，反例：$f(n)=1/n$。

若此时$f(n)=O(f(n)^2)=O(1/n^2)$，则存在正常量c和n_0，使得对所有$n\geq n_0$，有$0\leq 1/n\leq c/n^2$，即$c\geq n$。n可以足够大，则不存在满足条件的c。

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

正确。

$f(n)$与$g(n)$都是渐进非负函数，则存在正常量n_0，使得对所有$n\geq n_0$，$f(n),g(n)\geq 0$。

对于$c_1\max(f(n),g(n))\leq f(n)+g(n)\leq c_2\max(f(n),g(n))$，取$c_1=1,c_2=2$，满足不等式对于$n\geq n_0$成立。

c) $f(n)+O(f(n))=\Theta(f(n))$.

正确。

该等式表明，对任意函数$g(f(n))\in O(f(n))$，存在某个函数$h(f(n))\in\Theta(f(n))$，使得对所有n，有$f(n)+g(f(n))=h(f(n))$，记为$f+g=h$。

其中对所有$n\geq n_0$，存在正常量c_0，使得$0\leq g\leq c_0f$。

又由$f(n)$是渐进非负函数，则存在正常量n_1，使得对所有$n\geq n_1$，$f\geq 0$。

于是对于$c_1f\leq f+g\leq c_2f$，取$c_1=1,c_2=1+c_0$，有$f\leq f+g\leq(1+c_0)f$，满足不等式对于$n\geq\max(n_0,n_1)$成立。

d) if $f(n)=\Omega(g(n))$, then $g(n)=o(f(n))$.（注意是小o）

错误，反例：$f(n)=g(n)=n$。

此时满足$n=\Omega(n)$，但$\lim_{n\to\infty}\frac{g(n)}{f(n)}=1$。

2.时间复杂度

a) 证明 $\lg(n!)=\Theta(n\lg(n))$ (课本等式 3.19)，并证明$n!=\omega(2^n)$且$n!=o(n^n)$.

证明：

由$n!\le n^n$，有

$$\lg(n!)\le\lg(n^n)=n\lg n.$$

由Stirling公式：当$n\ge 1$时，$n!\ge(\frac{n}{e})^n$。于是有

$$\lg(n!)\ge\lg((\frac{n}{e})^n)=n(\lg n-\lg e).$$

令$c_0 n\lg n\le n(\lg n-\lg e)$，于是有

$$c_0\le\frac{\lg n-\lg e}{\lg n}=1-\frac{\lg e}{\lg n}$$

可以选择任何常量$c_0\in(0,0.37]$使得不等式对所有$n\ge 5$都成立。

因此通过选择$c_0=0.37,c_1=1,n_0=5$，对所有$n\ge n_0$，有

$$c_0 n\lg n\le\lg(n!)\le c_1 n\lg n$$

于是$\lg(n!)=\Theta(n\lg(n))$。

又由于

$$\lim_{n\to\infty}\frac{n!}{n^n}=\lim_{n\to\infty}\frac{\sqrt{2\pi n}(n/e)^n}{n^n}=\lim_{n\to\infty}\frac{\sqrt{2\pi n}}{e^n}=\lim_{n\to\infty}\frac{\sqrt{2\pi}}{2\sqrt{n}e^n}=0$$

$$\lim_{n\to\infty}\frac{n!}{2^n}=\lim_{n\to\infty}\frac{\sqrt{2\pi n}n^n}{(2e)^n}=\lim_{n\to\infty}e^{\ln\sqrt{2\pi n}+n(\ln n-\ln 2e)}=\infty$$

于是$n!=\omega(2^n)$且$n!=o(n^n)$。

b) 使用代入法证明 $T(n)=T(\lceil n/2\rceil)+1$**的解为**$O(lg(n))$.

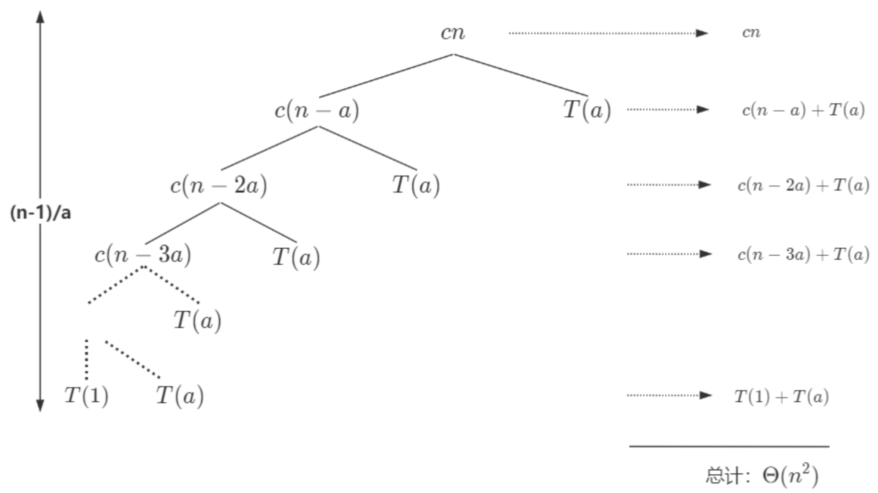
证明:

假设$0\le T(n)\le c\lg(n-d)$，d为非负常量，此上界对所有正数$m<n$都成立，有$0\le T(\lceil n/2\rceil)\le c\lg(\lceil n/2\rceil-d)$。代入递归式，有:

$$\begin{aligned}0\le T(n)&\le c\lg(\lceil n/2\rceil-d)+1\\&\le c\lg(n/2+1-d)+1\\&=c\lg(\frac{n+2-2d}{2})+1\\&=c\lg(n+2-2d)-c\lg 2+1\\&\le c\lg(n-d)\end{aligned}$$

此式对$d\ge 2, c\ge 1$成立。于是存在正常量$c\ge 1$，对所有正整数n有$0\le T(n)\le c\lg(n-d)\le c\lg(n)$，即$T(n)=O(\lg(n))$。

c) 对递归式 $T(n)=T(n-a)+T(a)+cn$,**利用递归树给出一个渐进紧确解,其中** $a\ge 1$**和**$c>0$**为常数.**



假设a可以被$n-1$整除，深度为i的结点对应规模为$n-ia$的子问题，因此当$n-ia=1$，即$i=\frac{n-1}{a}$时，子问题规模变为1(深度为0,1,...,$\frac{n-1}{a}$)。所以递归树有$\frac{n-1}{a}+1$层。注意到每一层的代价都为cn，所以总代价为:

$$T(n)=\sum_{i=0}^{\frac{n-1}{a}-1}c(n-ia)+\frac{n-1}{a}T(a)+T(1)=c\frac{n^2+an-1-a}{2a}+\frac{n-1}{a}T(a)+T(1)=\Theta(n^2)$$

于是推导出一个猜测：$T(n)=\Theta(n^2)$，下面用代入法来验证猜测。

假设存在正常量d_1使得$T(n)\le d_1n^2$对所有正数$m<n$都成立，于是有:

$$T(n)\le d_1(n-a)^2+d_1a^2+cn=d_1n^2-(2d_1an-2d_1a^2-cn)$$

令$d_1=(c+1)/2a$，只要$n \geq a(c+1)$则有：

$$T(n) \leq d_1 n^2 - (n-ac-a) \leq d_1 n^2$$

假设存在正常量d_2使得$T(n) \geq d_2 n^2$对所有正数$m<n$都成立，于是有：

$$T(n) \geq d_2(n-a)^2 + d_2 a^2 + cn = d_2 n^2 + 2d_2 a^2 + cn - 2d_2 an$$

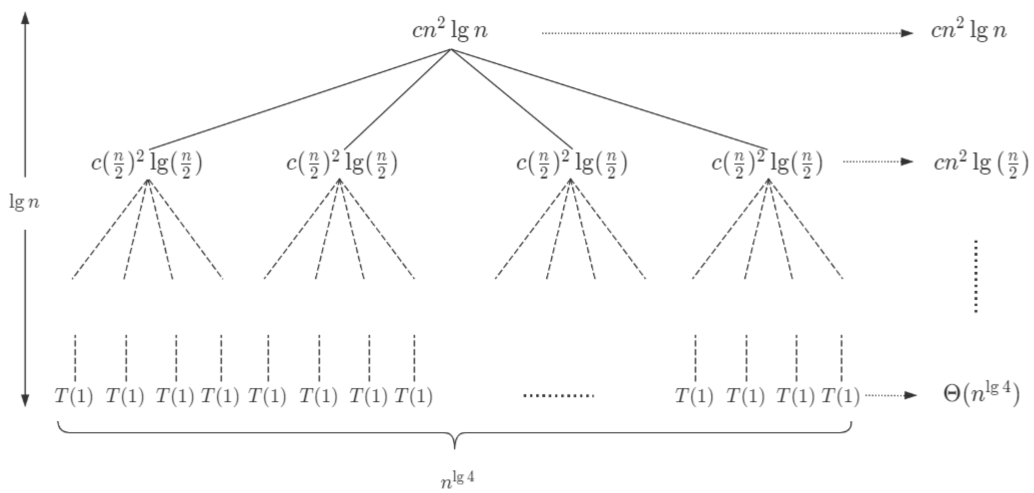
令$d_2 = c/2a$，有：

$$T(n) \geq d_2 n^2 + 2d_2 a^2 \geq d_2 n^2$$

于是有$T(n)=\Theta(n^2)$。

d) 主方法能应用于递归式 $T(n)=4T(n/2)+n^2\lg n$吗？请说明为什么可以或者为什么不可以. 给出这个递归式的一个渐进上界.

不可以，因为该递归式不符合主方法的任何一种情况。下面用递归树法求解。



深度为i的结点对应规模为$n/2^i$的子问题，因此当$n/2^i=1$，即$i=\lg n$时，子问题规模变为1。所以递归树有$\lg n+1$层。树的最底层深度为$\lg n$，有$4^{\lg n}=n^{lg4}=n^2$个结点，每个结点的代价为$T(1)$，该层总代价为$n^2T(1)$，即$\Theta(n^2)$。下面确定整棵树的代价：

$$
\begin{aligned}
T(n) &= \sum_{i=0}^{\lg n-1} cn^2 \lg(\frac{n}{2^i}) + \Theta(n^2) \\
&= cn^2 \sum_{i=0}^{\lg n-1} (\lg n - i) + \Theta(n^2) \\
&= cn^2(\lg^2 n - \sum_{i=0}^{\lg n-1} i) + \Theta(n^2) \\
&\leq cn^2 \lg^2 n + \Theta(n^2) \\
&= O(n^2 \lg^2 n)
\end{aligned}
$$

于是推导出一个猜测：$T(n) = O(n^2 \lg^2 n)$，下面用代入法来验证猜测。假设存在正常量c使得$T(n) \leq cn^2 \lg^2 n$对所有正数$m < n$都成立，于是有：

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

令$2c - 1 \geq c$，即$c \geq 1$，有：

$$T(n) = cn^2 \lg^2 n - ((2c-1)n^2 \lg n - cn^2) \leq cn^2 \lg^2 n$$

于是有$T(n) = O(n^2 \lg^2 n)$。

3.对下列递归式, 使用主方法求出渐进紧确解:

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

对于这个递推式，我们有$a = 2, b = 4, f(n) = \sqrt{n}$。因此$n^{\log_b a} = n^{\log_4 2} = \sqrt{n} = \Theta(\sqrt{n})$。由于$f(n) = \Theta(\sqrt{n})$，应用主方法的情况2，有$T(n) = \Theta(\sqrt{n} \lg n)$。

b) $T(n) = 2T(n/4) + n^2$

对于这个递推式，我们有$a = 2, b = 4, f(n) = n^2$。因此$n^{\log_b a} = n^{\log_4 2} = \sqrt{n} = \Theta(\sqrt{n})$。由于$f(n) = \Omega(n^{\log_4 2+\epsilon})$，其中$\epsilon = 1.5$。当$n$足够大时，对于$c = 1/4$，$af(n/b) = n^2/8 \leq n^2/4 = cf(n)$。因此由主方法的情况3，有$T(n) = \Theta(n^2)$。

4.考虑以下查找问题：

输入：n个数的一个序列$A = a_1, a_2, \dots, a_n$和一个值 v .
输出：下标 i 使得$v = A[i]$或者当v不在 A 中出现时,v为特殊值NIL.

a) 写出*线性查找*的伪代码, 它扫描整个序列来查找v. 使用一个Loop Invariant (循环不变式) 来证明你的算法是正确的.

```
LINEAR-SEARCH(A,v)
//假设A中元素有重复
    flag = 0
    for i = 1 to A.length
        if A[i] == v
            flag = 1
            print i
    if flag == 0
        print  NIL
```

初始化：首先证明在第一次循环迭代前（当$i = 1$时），循环不变式成立。这时flag初始化为0，表示还未找到下标 i 使得$v = A[i]$，循环不变式显然成立。

保持：每次循环后i增1，只要有一个下标 i 使得$v = A[i]$，flag会标记为1，且输出下标i，无论接下来是否还有满足条件的下标，flag=1保持不变。反之，若在$i \le A.length$时，没有找到满足条件的下标，则flag=0保持不变。

终止：当i大于$A.length$即遍历完整个序列，循环结束。这时若flag未发生变化即flag=0，表示没有找到满足条件的下标，输出NIL。若falg发生变化即flag=1，表示找到了至少一个满足条件的下标。

b) 假定 v 等可能的为数组中的任意元素，平均需要检查序列的多少元素？最坏情况又如何呢？用 Θ记号给出线性查找的平均情况和最坏运行时间.

平均情况：$\Theta(n)$

最坏情况：$\Theta(n)$

5.堆排序：

对于一个按升序排列的包含n个元素的有序数组A来说，$\mathrm{HEAPSORT}$的时间复杂度是多少？如果A是降序的呢？请简要分析并给出结果.

降序：建堆时，初始堆已经是最大堆，进行$O(n)$次 `MAX-HEAPIFY(A,i)` 的调用，每次的时间复杂度为$O(1)$，所以建堆的时间复杂度为$O(n)$。排序时，$n-1$次调用 `MAX-HEAPIFY(A,i)` ，每次$A[1]$与$A[i]$交换后， `MAX-HEAPIFY(A,i)` 都要再进行$\lg n$次，所以这n次循环的时间复杂度为$O(n\lg n)$。总时间复杂度为$O(n\lg n)$。

升序：建堆时，进行$O(n)$次 `MAX-HEAPIFY(A,i)` 的调用，建堆的时间复杂度为$O(n)$。排序时，$n-1$次调用 `MAX-HEAPIFY(A,i)` ，每次$A[1]$与$A[i]$交换后， `MAX-HEAPIFY(A,i)` 都要再进行$\lg n$次，所以这n次循环的时间复杂度为$O(n\lg n)$。总时间复杂度为$O(n\lg n)$。

6.快速排序：

1. **假设快速排序的每一层所做的划分比例都是$1-\alpha:\alpha$，其中$0<\alpha\leq 1/2$且是一个常数. 试证明：在相应的递归树中，叶结点的最小深度大约是$-lgn/lg\alpha$，最大深度大约是$-lgn/lg(1-\alpha)$(无需考虑舍入问题).**

   为求最小深度，每一次划分选择最小的一部分，每次迭代剩余元素个数乘以α，迭代次数即树的深度。直到迭代k_1次剩下一个元素，有$\alpha^{k_1}n=1$即$k_1=-lgn/lg\alpha$。

   求最大深度同理，每一次划分选择最大的一部分，每次迭代剩余元素个数乘以$1-\alpha$，有（$1-\alpha$）$^{k_2}n=1$即$k_2=-lgn/lg(1-\alpha)$。

2. **试证明：在一个随机输入数组上，对于任何常数$0<\alpha\leq 1/2$，$\mathrm{PARTITION}$产生比$1-\alpha:\alpha$更平衡的划分的概率约为$1-2\alpha$.**

   证明：

   假设数组长度为n，$1-\alpha:\alpha$的划分将数组划分为两部分，长度为αn和$(1-\alpha)n$。则划分点在数组从前往后数第αn处或者第$(1-\alpha)n$处。要产生更好的划分，则划分点应在第αn处至第$(1-\alpha)n$处之间。考虑到划分点的选择等可能，服从0至n的均匀分布，有：$P(\alpha n\leq x\leq(1-\alpha)n)=((1-\alpha)n-\alpha n)/n=1-2\alpha$。