# Homework 1

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

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

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

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

**2.** 时间复杂度

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

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

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 $a \geq 1$ 和 $c > 0$ 为常数.

d) 主方法能应用于递归式 $T(n) = 4T(n/2) + n^2 lgn$ 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.

**3.** 对下列递归式, 使用主方法求出渐进紧确解:

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

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

**4.** 考虑以下查找问题:
**输入:** $n$ 个数的一个序列 $A = a_1, a_2, ..., a_n$ 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.

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 $\Theta$ 记号给出线性查找的平均情况和最坏运行时间.

**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，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.

**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 $-\lg n/\lg\alpha$，最大深度大约是 $-\lg n/\lg(1-\alpha)$ (无需考虑舍入问题).
2. 试证明：在一个随机输入数组上，对于任何常数 $0 < \alpha \leq 1/2$，Partition 产生比 $1-\alpha:\alpha$ 更平衡的划分的概率约为 $1-2\alpha$.

1. (a) 错误 设 $f(n)=\frac{1}{n}$

$\forall c, n_0 \quad \exists n > \max(c, n_0) \quad \frac{1}{n} > \frac{c}{n^2} > 0$ 故 $f(n) \neq O(f(n)^2)$

(b) 正确

$\max(f(n), g(n)) \le f(n)+g(n) \le 2\max(f(n), g(n))$

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

(c) 正确 设 $O(f(n)) = g(n)$

则 $0 \le g(n) \le cf(n)$

$f(n) \le f(n) + O(f(n)) \le (1+c)f(n)$

故 $f(n) + O(f(n)) = \Theta(f(n))$

(d) 错误 设 $f(n) = g(n) = n$

$f(n) = \Omega(g(n))$

而 $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 1 \neq 0$ 故 $g(n) \neq o(f(n))$

2 a) 由 stirling 公式: $n! = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n e^{\frac{\theta_n}{12n}}$ $\theta_n \in (0,1)$

$\lg n! = \lg\sqrt{2\pi} + \frac{1}{2}\lg n + n\lg n - n\lg e + \frac{\theta_n}{12n}\lg e$

n 足够大时 $n\lg n - n\lg e < \lg n! < n\lg n$ $\quad n\lg n - n\lg e = \Omega(n\lg n)$

故 $\lg n! = \Theta(n\lg n)$

$\lim_{n\to\infty} \frac{n!}{2^n} = \frac{\sqrt{2\pi n}\left(\frac{n}{e}\right)^n e^{\frac{\theta_n}{12n}}}{2^n} = \infty$ $\quad \lim_{n\to\infty} \frac{n!}{n^n} = \frac{\sqrt{2\pi n}\left(\frac{n}{e}\right)^n e^{\frac{\theta_n}{12n}}}{n^n} = 0$ 故得证.

2b) $T(n)=T(\lceil n/2\rceil)+1$  $T(n)=O(\lg n)$

假设对$\forall m<n$都成立 特别对$m=\lceil n/2\rceil$

$$T(n)\le c\lg\left\lceil\frac{n}{2}\right\rceil+1\le c\lg\left(\frac{n}{2}+1\right)+1$$

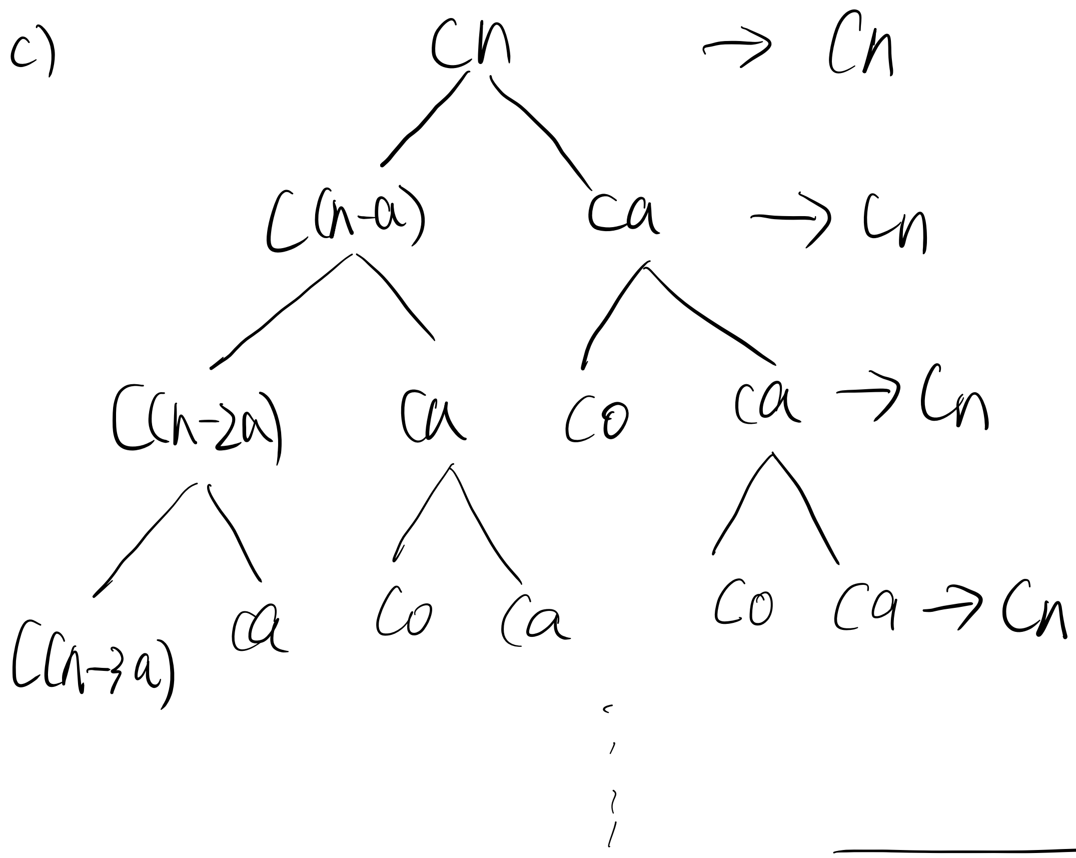
(不妨设 $n>6$) $\le c\lg\frac{2}{3}n+1=c\lg n-c\lg\frac{3}{2}+1$

(c足够大时) $\le c\lg n$

对于边界 只要c取足够大都能满足

故上式的解为 $O(\lg n)$

2 c)




共计$\lceil n/a\rceil$

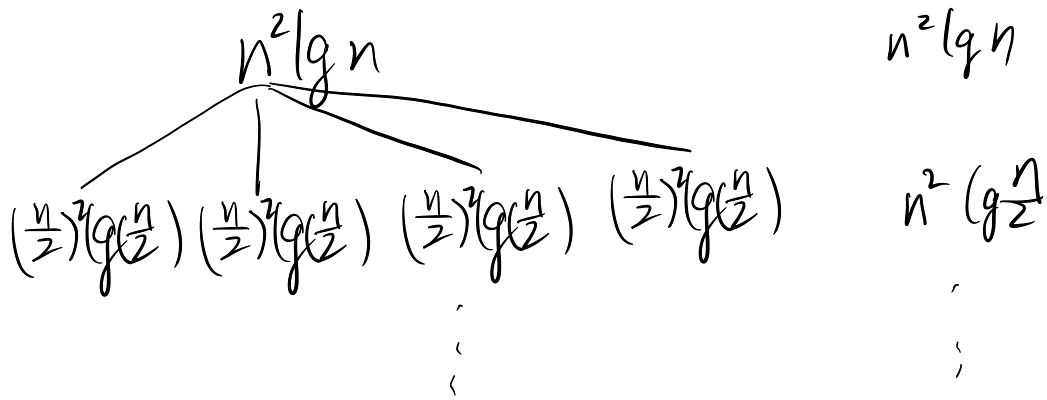
故 $\frac{n}{a}\cdot cn\le T(n)\le\left(\frac{n}{a}+1\right)cn$  故 $T(n)=\Theta(n^2)$

2d) $T(n) = 4T(\frac{n}{2}) + n^2\lg n$

$\log_b a = \log_2 4 = 2$ 而 $\forall \varepsilon > 0$ $\frac{f(n)}{n^{\log_b a}}$ 渐近小于 $n^\varepsilon$

故不能用主方法

构造递归树：




故 $T(n) = n^2(\underbrace{\lg n + \lg\frac{n}{2} + \cdots}_{\log_2 n})$

$= n^2\log_2 n\cdot\lg n + n^2(\lg 1 + \lg\frac{1}{2} + \lg\frac{1}{4} + \cdots)$

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

$= \frac{n^2(\lg n)^2}{\lg 2} - n^2\frac{\log_2 n(\log_2 n+1)}{2}\lg 2$

$= O(n^2\lg^2 n)$

3. a) $T(n) = 2T(\frac{n}{4}) + \sqrt{n}$

$a=2$ $b=4$ $\log_b a = \frac{1}{2}$ $f(n) = \sqrt{n} = \Theta(n^{\frac{1}{2}})$

故 $T(n) = \Theta(\sqrt{n}\lg n)$

b) $T(n) = 2T(\frac{n}{4}) + n^2$

$\log_b a = \frac{1}{2}$　　$\varepsilon = \frac{3}{2}$ $f(n) = n^2 = n^{\frac{1}{2}+\frac{3}{2}} = n^{\log_b a + \varepsilon}$

$= \Omega(n^{\log_b a + \varepsilon})$

而 $2(\frac{n}{4})^2 = \frac{n^2}{8} \le \frac{2}{8}n^2$ 故取 $c = \frac{2}{8} < 1$

满足主方法的 3

故 $T(n) = \Theta(n^2)$

4. a) SEARCH ( A, v )

```
for i = 1 to A.length
    if A[i] == v
        return i
return NIL
```

初始化：$i=1$ 时 查找 v 若 $A[1]==v$ 则返回 1

保持：每次循环 $i=i+1$ 比较 $A[i]$ 与 v

若 $A[i]==v$ 则返回 i 与 $i=1$ 的查找方式相同

终止：当找到 i $A[i]==v$ 时会终止返回 i

若 $i > A.length$ 时 说明未找到 $A[i]==v$

返回 NIL 算法正确

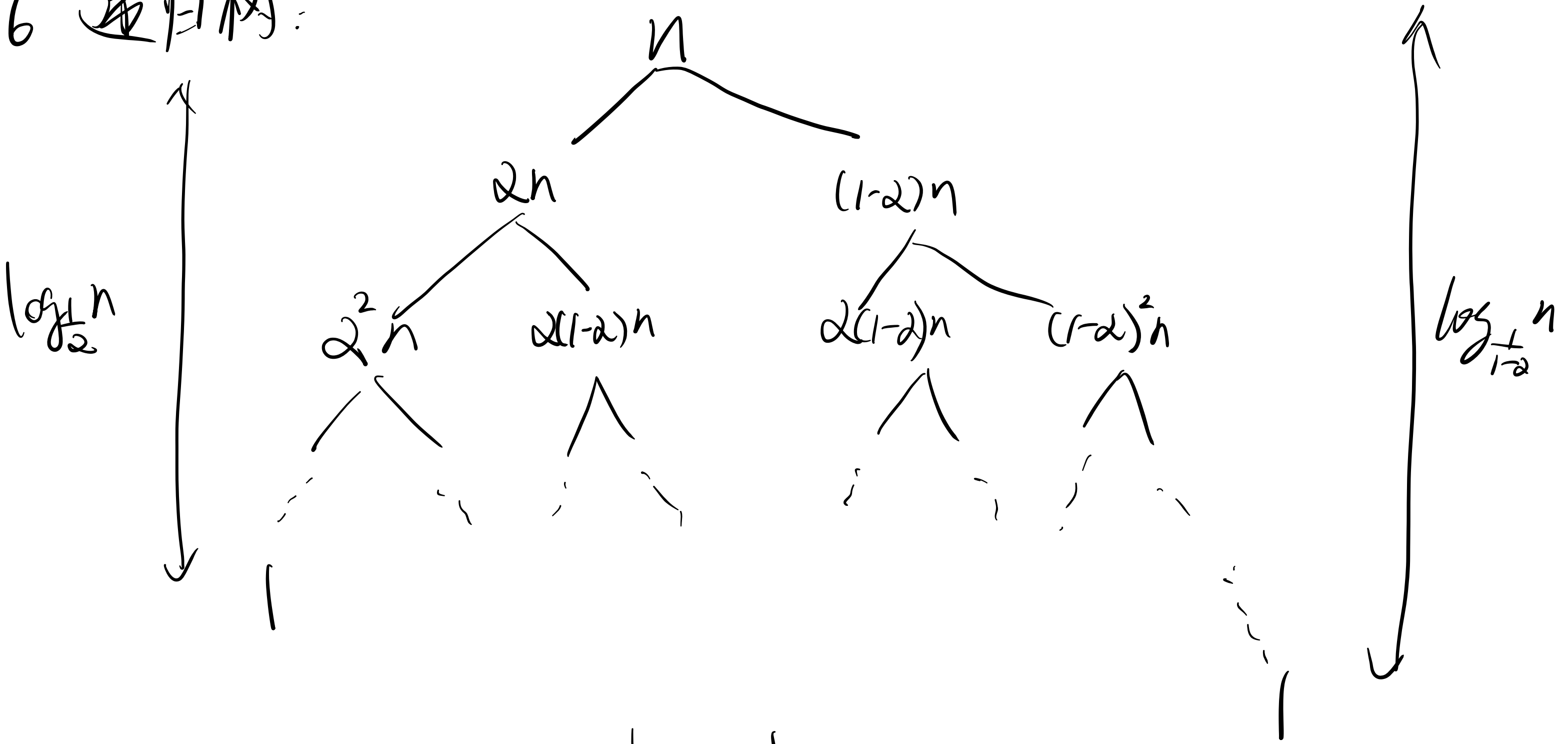
b) AVE：$\frac{1}{n}(1+2+\cdots+n) = \frac{1}{n} \cdot \frac{n(n+1)}{2} = \frac{n+1}{2} = \Theta(n)$

WORST：v 不在 A 中 要检查 n 个 故为 $\Theta(n)$

5. 无论升序还是降序都是 $n\lg n$

因为堆排序过程中尾部数字换到顶部还是要下沉，比较的时间仍然是 $n\lg n$

6 递归树:




不考虑舍入时 设 $n$ 是 $\frac{1}{\alpha}$ 与 $\frac{1}{1-\alpha}$ 的整数次幂 因为 $\alpha \le 1-\alpha$

故：最左侧深度为 $\log_{\frac{1}{\alpha}} n = -\frac{\lg n}{\lg \alpha}$ 为最小深度

最右侧深度为 $\log_{\frac{1}{1-\alpha}} n = -\frac{\lg n}{\lg(1-\alpha)}$ 为最大深度

2. 当划分点 $k$ 满足 $\alpha n < k < (1-\alpha)n$ 时 更平衡

故更平衡划分的概率为 $\frac{(1-\alpha)n-\alpha n-1}{n} \approx 1-2\alpha$

得证.