算法导论　第二周作业　9 月 24 日　周四

PB18151866 龚小航

1.1 考虑以下查找问题:

输入: n 个数的一个序列 $A = a_1, a_2, \ldots, a_n$ 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(查找失败), v 为特殊值 NIL.

$(\boldsymbol{a})$. 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.

$(\boldsymbol{b})$. 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.

解: 对这两个问题分别分析:

$(\boldsymbol{a})$ 先写出线性查找的伪代码, 即是从头到尾顺序查找:

```
SEARCH (A, v) :
    for i = 1 to A.length
        if A[i] == v
            return i
    return NIL
```

再为此构造循环不变式, 此处将循环不变式表示为 $A[1,\cdots\cdots,i-1]$, 下证循环不变式的三条属性:

- 初始化:

  初始时, $i = 1$, 此时子数组为空, 由于子数组没有任何数据, 即为找不到任何匹配 v 的下标 i, 此时循环不变式成立。

- 保持:

  在每一步, 已经知道 $A[1,\cdots\cdots i-1]$ 中不包含 v , 这时比较 v 和 $A[i]$. 若它们是相同的则返回 i, 这是我们希望看到的正确结果; 若不相等则程序继续向下执行。上述算法保证 $A[1,\cdots\cdots i-1]$ 中不包含 v 且 $A[i]$ 与 v 不同, 这一步保持了循环不变式。

- 终止:

  当 $i == A.length + 1$ 时算法必然终止, 由上两步的推导, 可以知道 A 中的全部元素都已经检查过, 且 v 不在 A 内, 此时返回特殊值 NIL。因此算法正确。

$(\boldsymbol{b})$ 若 v 等可能的出现在数组的各个位置中, 计算其查找性能:　令 $A.length = n$

$$\text{平均查找元素数} = \frac{\sum_{i=1}^{n} i}{n} = \frac{n+1}{2}$$

$$\text{最坏情况查找元素数} = n$$

两种情况的时间复杂度都可以表示为 $\Theta(n)$

1.2 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 $f(n)+g(n) = \Theta\left(max\left(f(n),g(n)\right)\right)$.

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

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

解：分析并结合举例说明：

(***a***)： 将其描述展开，$f(n) = O(f(n)^2)$ 即为：

存在正常数 c 和 n_0，使对所有 $n \geq n_0$，都有 $0 \leq f(n) \leq c \cdot f^2(n)$

注意先取 c, n_0 ，再取 n，因此 c 的取值不能和 n 有关。

- 取反例，令 $f(n) = 1/n$，由于 n 可以取到无穷大，只需令 $n > c$，即有 $f^2(n) = c\frac{1}{n}\frac{1}{n} < \frac{1}{n} = f(n)$

因此这个等式是不一定成立的。

(***b***)： 将其描述展开，$f(n)+g(n) = \Theta\left(\max\left(f(n),g(n)\right)\right)$ 即为：

存在正常数 c_1, c_2 和 n_0，使对所有 $n \geq n_0$，都有：

$$0 \leq c_1\max\left(f(n),g(n)\right) \leq f(n)+g(n) \leq c_2\max\left(f(n),g(n)\right)$$

- 由于 $f(n), g(n)$ 是渐进非负函数，因此 $\exists n_0$ 使得当 $n \geq n_0$ 时有 $f(n), g(n) \geq 0$ 成立；
- 显然有： $f(n)+g(n) \leq 2\max\left(f(n),g(n)\right)$，因此取 $c_2 \geq 2$ 即可；
- 另一边，令 $0 \leq c_1 \leq 1$，就可满足 $0 \leq c_1\max\left(f(n),g(n)\right) \leq f(n)+g(n)$；

综上两边，可得原等式是成立的

(***c***)：$f(n) + O(f(n)) = \Theta\left(f(n)\right)$ ，将左边展开成一个整体函数，先将 $O\left(f(n)\right)$ 展开成 $g(n)$：

存在正常数 c 和 n_0，使对所有 $n \geq n_0$，都有 $0 \leq f(n) \leq c \cdot g(n)$

再把 Θ 按定义展开，即：

存在正常数 c_1, c_2 和 n_0，使对所有 $n \geq n_0$，都有：

$$0 \leq c_1 f(n) \leq f(n)+g(n) \leq c_2 f(n)$$

这显然是不一定成立的。比如可以取 $g(n) = nf(n)$，当 $n \to \infty$ 时显然右侧 $\leq$ 不等式不可能成立。

因此这个等式不一定成立。

(***d***)： $f(n) = \Omega\left(g(n)\right) \implies f(n) = o\left(g(n)\right)$

将其前提描述展开，$f(n) = \Omega\left(g(n)\right)$ 即为：

存在正常数 c 和 n_0，使对所有 $n \geq n_0$，都有 $0 \leq c \cdot g(n) \leq f(n)$

再把结论描述展开，写成：

对任意的正常数 c_1，存在正常数n_1，使对所有 $n \geq n_1$，都有 $0 \leq f(n) \leq c_1 \cdot g(n)$

这显然是不一定成立的，例如令 $n \to \infty$，按前提必有 $0 \leq c \cdot g(n) \leq f(n)$，此时只要取 $c_1 = c$，可知 $c_1 \cdot g(n) \leq f(n)$，显然只有取等号条件时才有可能成立，大部分情况下这两者不相等。

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

解：　将其描述展开，$\lg(n!) = \Theta(n \cdot \lg(n))$ 即为：　　（ lg 在计算机中表示以 2 为底的对数 ）

存在正常数 c_1, c_2 和 n_0，使对所有 $n \geq n_0$，都有：$0 \leq c_1 n\lg(n) \leq \lg(n!) \leq c_2 n\lg(n)$

- 由于 $\lg(n!)$ 显然是渐进非负函数，因此 $\exists n_0$ 使得当 $n \geq n_0$ 时有 $\lg(n!) \geq 0$ 成立；
- $\lg n! = \sum_{i=1}^{n} \lg i = \sum_{i=2}^{n} \lg i \geq \sum_{i=2}^{n} \lg \sqrt{n} = \frac{1}{2} n \lg n$只需取 $0 \leq c_1 \leq 1/2$，即可满足左不等式；
- 另一边，$\lg n! = \sum_{i=1}^{n} \lg i \leq \sum_{i=1}^{n} \lg n = n \lg n$，只需令 $c_2 \geq 1$ 即可满足右不等式

综上两边，可得原等式是成立的，即 $\lg(n!) = \Theta(n \cdot \lg(n))$

对于这个问题，也可以直接利用斯特林公式，即把 $n!$ 展开即可得：

$$\lg n! \approx \lg\left(\sqrt{2\pi n}\left(\frac{n}{e}\right)^n\right) \approx \lg n^n = n \lg n$$

立刻就可得到待证结论。

证明 $n! = \omega(2^n)$，将其描述展开：

对任意的正常数 c，存在正常数n_0，使对所有 $n \geq n_0$，都有 $0 \leq c \cdot 2^n \leq n!$

- 由于 $n!$ 显然是渐进非负函数，因此 $\exists n_0$ 使得当 $n \geq n_0$ 时有 $n! \geq 0$ 成立；
- 直接对 $n!$ 和 2^n 相除进行比较即可：

$$\frac{n!}{2^n} = \frac{n(n-1)(n-2)\cdots\cdots * 2 * 1}{2 * 2 * 2 * 2 \cdots\cdots * 2} = \frac{n}{2} + \frac{n-1}{2} + \cdots\cdots + \frac{1}{2}$$

对于一个任意给定的正常数 c，要使上式大于 c 成立：

$$\frac{n!}{2^n} = \frac{n}{2} + \frac{n-1}{2} + \cdots\cdots + \frac{1}{2} \geq \frac{1}{2} + \frac{1}{2} + \cdots\cdots + \frac{1}{2} = \frac{n}{2} > c$$

即在给定任意正常数 c 之后，取 $n_0 = 2c$ 即有 $0 \leq c \cdot 2^n \leq n!$

综上，$n! = \omega(2^n)$ 成立。

再证明 $n! = o(n^n)$，将其描述展开：

对任意的正常数 c，存在正常数n_0，使对所有 $n \geq n_0$，都有 $0 \leq n! \leq c \cdot n^n$

- 由于 $n!$ 显然是渐进非负函数，因此 $\exists n_0$ 使得当 $n \geq n_0$ 时有 $n! \geq 0$ 成立；
- 直接对 $n!$ 和 n^n 相除进行比较即可：

$$\frac{n!}{n^n} = \frac{n(n-1)(n-2)\cdots\cdots * 1}{n * n * n * \cdots\cdots * n} = \prod_{i=1}^{n-1}\left(1 - \frac{i}{n}\right) < 1/n$$

对于一个任意给定的正常数 c，要使上式小于 c 成立：

$$\frac{n!}{n^n} = \prod_{i=0}^{n-1}\left(1 - \frac{i}{n}\right) < \frac{1}{n} < c$$

即简单的取 $n > 1/c$ 就能满足条件

因此原等式是成立的。

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

解：使用代入法需要证明的是：

存在正常数 c 和 n_0，使对所有 $n \geq n_0$，都有：$T(n) \leq c\lg n$

利用数学归纳法证明：

- **归纳基础：** $n = 1\ or\ n = 2$ 时：令 $T(1) = 1$,无论 c 取何值都无法满足 $T(1) \leq c \lg 1 = 0$. 这里可以保留麻烦的边界条件 $T(1) = 1$，但将其从归纳证明中移除。此处以 $T(2)$ 作为归纳证明的归纳基础。

  $n = 2$ 时，$T(2) = T(1) + 1 = 2 \leq c \cdot \lg 2 = c$ 可以轻易满足，只要取 $c \geq 2$ 即可。
- **归纳假设：** 当 $n \geq 3$ 时，假设 $T(k) \leq c \cdot \lg k$ 对所有的 $k = 1,2,3,\cdots\cdots, n-1$ 均成立
- **归纳递推：** 当 $n \geq 3$ 时，再证明 $T(n) \leq c \cdot \lg n$ 成立：

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

再取 c 使得 $c \cdot \lg\frac{n+1}{2} + 1 \leq c\lg n$ 成立即可：

$$c \cdot \lg\frac{n+1}{2} + 1 \leq c\lg n \quad \Leftrightarrow \quad c\lg n - c\lg\frac{n+1}{2} \geq 1 \quad \Leftrightarrow \quad c\lg\frac{2n}{n+1} \geq 1$$

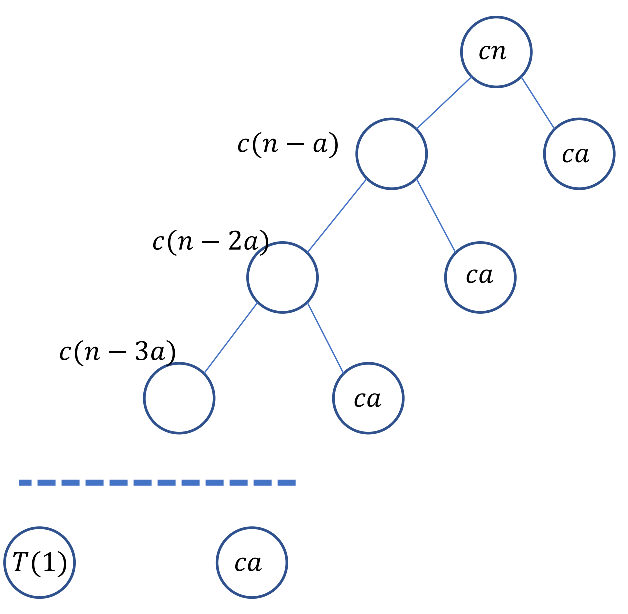
$$\Leftrightarrow \quad c \geq \frac{1}{\lg\dfrac{2}{1+\dfrac{1}{n}}} \quad \text{记作 } f(n)$$

当 $n \geq 3$ 时，$f(n)$ 单调递减，因此只要取 $c > 1/(\lg 3 - 1)$ 即可使 $T(n) \leq c\lg n$ 成立

综上，c 需要满足的条件为 $c \geq 2$,因此原猜测是成立的，$T(n) = T(\lceil n/2 \rceil) + 1$ 的解为 $O(\lg n)$

1.5　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. ($a \geq 1, c > 0$ 为常数.)

解：做出其递归树：



根据递归树，可知：

$$T(n) = \sum_{i=0}^{n\backslash a} c(n - ia) + \left(\frac{n}{a}\right) ca$$

$$= \frac{cn^2}{a} - ca\sum_{i=0}^{n\backslash a} i + cn$$

$$= \frac{cn^2}{2a} + \frac{cn}{2}$$

$$= O(n^2)$$

1.6　对下列递归式，使用主方法求出渐近紧确解:

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

$(\boldsymbol{b})$.　$T(n) = 2T(n/4) + n^2$.

解：分别对以下两种情况分析：

$(\boldsymbol{a})$　$T(n) = 2T\left(\frac{n}{4}\right) + \sqrt{n} \implies a = 2,\ b = 4,\ f(n) = \sqrt{n} \implies n^{\log_{\mathrm{b}} a} = n^{\frac{1}{2}} \implies f(n) = \Theta(n^{\log_b a})$

对应于情况二，因此有 $T(n) = \Theta\left(n^{\log_b a} \lg n\right) = \Theta\left(\sqrt{n} \lg n\right)$

$(\boldsymbol{b})$　$T(n) = 2T\left(\frac{n}{4}\right) + n^2 \implies a = 2,\ b = 4, f(n) = n^2 \implies n^{\log_b a} = \sqrt{n} \implies f(n) = \Omega\left(n^{\log_b a + \frac{3}{2}}\right)$

判断是否可用于情况三，此时已经求出 $\varepsilon = 3/2$ ，接下来令 $n \to \infty$:

$$af\left(\frac{n}{b}\right) = a\frac{n^2}{b^2} = \frac{n^2}{8}; \quad cf(n) = cn^2$$

令 $af(n/b) \le cf(n)$，可得 $1/8 \le c < 1$

因此只要取 $1/8 \le c < 1$ 就可以满足情况三的使用条件，因此这个递归式满足情况三。

所以它的渐近紧确解为：

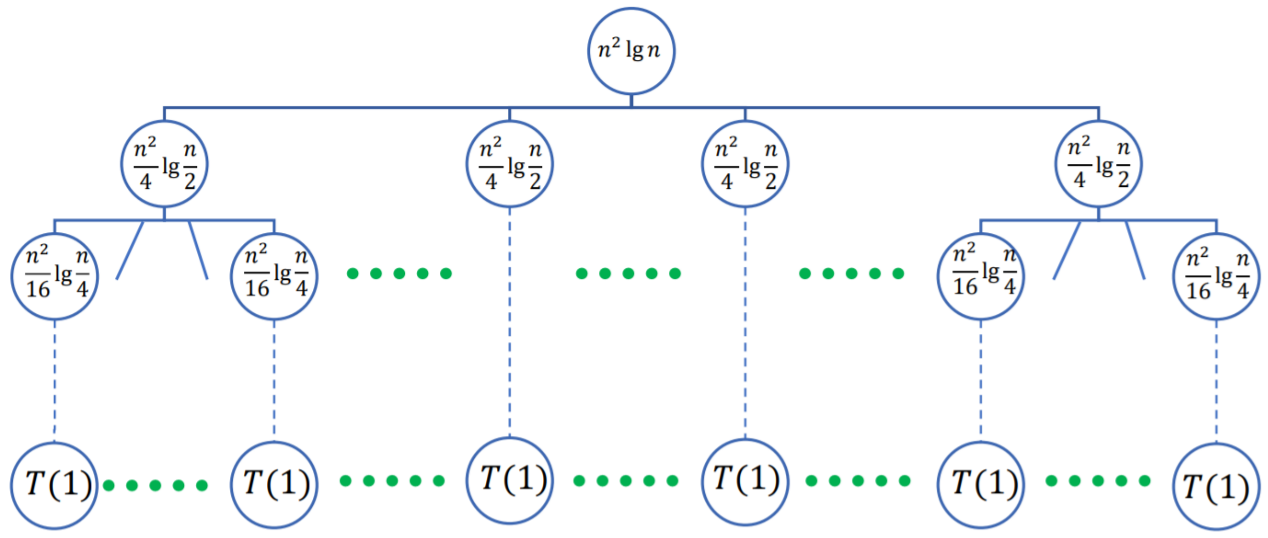
$$T(n) = \Theta\left(f(n)\right) = \Theta(n^2)$$

1.7 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.

解：$T(n) = 4T\left(\frac{n}{2}\right) + n^2\lg n \quad\Rightarrow\quad a=4, b=2, f(n)=n^2\lg n \quad\Rightarrow\quad n^{\log_b a} = n^2$

此时 $f(n)=n^2\lg n$ 渐近大于 $n^{\log_b a}$，但不是多项式意义的。对比主定理的情况三，不存在 $\varepsilon > 0$ 使得 $f(n)=\Omega(n^{\log_b a+\varepsilon})$

所以这个递归式不能用主方法求解。只能作出其递归树，求解其渐近上界：



在递归树中，深度为 i 的节点对应规模为 $n/2^i$ 的子问题，随着 i 的增长，问题规模会最终变为 1. 这里利用了一个“不精确的假设”，即令 n 是 2 的幂，这样所有子问题的规模都是正整数。此时树高 $h = \lg n$. 每一层的节点数是上一层的 4 倍，因此深度为 i 的节点数为 4^i。因此除去叶节点之外，深度为 i 的所有节点代价和为

$$4^i\cdot\left(\frac{n}{2^i}\right)\lg\left(\frac{n}{2^i}\right) = n^2\lg\left(\frac{n}{2^i}\right) = n^2(\lg n - i)$$

此外，所有叶节点的代价和为 $\Theta(n^2)$

再将每一层的代价相加，即可得：

$$T(n) = \sum_{i=0}^{\lg n-1}[n^2(\lg n - i)] + \Theta(n^2) = \frac{1}{2}n^2\lg^2 n + \Theta(n^2) = O(n^2\lg^2 n)$$

最后用代入法证明这个结果，即需要证明：

存在正常数 c, n_0，使对所有 $n \ge n_0$，都有 $0 \le T(n) \le c\cdot n^2\lg^2 n$ 成立

利用数学归纳法证明：

- **归纳基础：** $n=1$ *or* $n=2$ 时：令 $T(1)=1$,无论 c 取何值都无法满足 $T(1)\le c1^2\lg^2 1 = 0$. 这里可以保留麻烦的边界条件 $T(1)=1$，但将其从归纳证明中移除。以 $T(2)$ 作为归纳证明的归纳基础。
  $n=2$ 时，$T(2)=4T(1)+1^2\lg^2 1 = 4 \le c\cdot 2^2\lg^2 2 = 4c$ 可以轻易满足，只要取 $c\ge 1$ 即可。
- **归纳假设：** 当 $n\ge 3$ 时，假设 $T(k)\le c\cdot k^2\lg^2 k$ 对所有的 $k = 1,2,3,\cdots\cdots,n-1$ 均成立
- **归纳递推：** 当 $n\ge 3$ 时，再证明 $T(n)\le c\cdot n^2\lg^2 n$ 成立：

$$T(n) = 4T\left(\frac{n}{2}\right) + n^2\lg n \le 4c\cdot\left(\frac{n}{2}\right)^2\lg^2\left(\frac{n}{2}\right) + n^2\lg n = cn^2(\lg n - 1)^2 + n^2\lg n$$

再取 c 使得 $cn^2(\lg n-1)^2 + n^2\lg n \le cn^2\lg^2 n$ 成立即可：

$$cn^2(\lg n-1)^2 + n^2\lg n \le cn^2\lg^2 n \quad\Leftrightarrow\quad 2c\lg n - c \ge \lg n$$

$$\Leftrightarrow\quad c \ge \frac{1}{2-\frac{1}{\lg n}} \quad \text{记作 } f(n)$$

当 $n\ge 3$ 时，$f(n)$ 单调递减，因此只要取 $c > 1/(2-1/\lg 3)$ 即可使 $T(n)\le c\lg n$ 成立

综上，c 需要满足的条件为 $c\ge 1$,因此原猜测是成立的，$T(n) = 4T\left(\frac{n}{2}\right) + n^2\lg n$ 的解为 $O(n^2\lg^2 n)$