毛九弢-221900175-A1算法基础作业

1 算法正确性证明

1.1 辗转相除法

证明欧几里得算法(辗转相除法求最大公约数)的正确性, 以下为其伪代码 (采用类python格式):

```python
def gcd(a, b):
    """
    a, b are both integer, a >= 0, b >= 0
    """
    if a < b:
        return gcd(b, a)
    if b == 0:
        return a
    return gcd(b, a % b)
```

我们拓展 最大公约数的定义

认为 0与任何数的最大公约数为该数

我们称 第 i 次 调用的gcd(a,b) 中的a,b 为 a_i 和 b_i;

（一）先证明 对于所有存在的 a_i b_i （这里后文好像没有使用）

$$\text{有}\ (i > 1) \Rightarrow a_i \geq b_i$$

我们对 i 进行 归纳:

① 证 $i=2$ $a_2 > b_2$

if 1.1 $a_1 < b_1 \Rightarrow a_2 = b_1$ $b_2 = a_1$ $a_2 > b_2$

elif 1.2 $b_1 = 0 \Rightarrow$ 不存在 a_2, b_2 （此处不讨论）

else 1.3 $a_1 \geq b_1 \Rightarrow a_2 = b_1$ $b_2 = a_1 \% b_1 < b_1 \Rightarrow a_2 > b_2$

② 证 $i=k>1$ 成立时 $i=k+1$ 成立, 由于 $i=k$ 成立, 故 $a_k > b_k$

if 1.1: $b_k = 0 \Rightarrow$ 不存在 a_{k+1}, b_{k+1} （算法终止, 也不存在 a_j, b_j $(j>k)$）

else 1.2 $a_{k+1} = b_k$ $b_{k+1} = a_k \% b_k < b_k \Rightarrow a_{k+1} > b_{k+1}$

③ 归纳完成

（二）对算法的终止性证明:

对于 任意 第i次调用, 有 三种情况:

① $a_i < b_i$: $b_{i+1} = a_i < b_i = a_{i+1}$ 有 $b_{i+1} < b_i$

② $a_i \geq b_i$ && $b_i = 0$: b_{i+1}不存在，即算法终止;

③ $a_i \geq b_i$ && $b_i \neq 0$: $b_{i+1} = a_i \bmod b_i < b_i$

故而 要么 $b_{i+1} < b_i$ 要么 算法终止

故，在算法未终止时（即调用过程中）有不变式:

$$b_{i+1} < b_i$$

又由于 gcd函数中 运用的所有运算（赋值运算和模运算） 在自然数域内 封闭;

由 良序原理 得: 一定$\exists\, n \in N^*, s.t.\ b_n = 0$; 即第n次调用后 算法终止。

进而 不妨设经历n次调用gcd函数后，算法终止，并假设设 $\mathrm{rgcd}\,(a_n, b_n) = \mathrm{u}$;

由于递归调用， $\gcd(a_i, b_i) = \gcd(a_{i+1}, b_{i+1})$ 等式显然成立

（三）我们证明 调用中的不变式:

由于递归调用，$gcd(a_i, b_i) = gcd(a_{i+1}, b_{i+1})$ 等式显然成立

（三）我们证明 调用中的不变式：

$$\mathrm{rgcd}\left(a_i, b_i\right) = \mathrm{rgcd}\left(a_{i+1}, b_{i+1}\right) \quad (i < n, \text{否则} i+1 \text{无意义})$$

注意，这里我们使用 gcd 表示算法中的函数，rgcd表示两数的最大公约数

假设 $\mathrm{rgcd}\left(a_i, b_i\right) = c_i$，$\mathrm{rgcd}\left(a_{i+1}, b_{i+1}\right) = c_{i+1}$

if ① $a_i < b_i$，$a_{i+1} = b_i$，$b_{i+1} = a_i$，由最大公约数的对称性

$\therefore rgcd(a_i, b_i) = rgcd(b_i, a_i) = rgcd(a_{i+1}, b_{i+1})$

elif ② $b_i = 0$（在此不做讨论）

else ③ $a_{i+1} = b_i$，$b_{i+1} = a_i \bmod b_i$ 设 $a_i = k_{a_i} c_i$ $b_i = k_{b_i} c_i$

由于此时 $a_i \geq b_i$ 故 $k_{a_i} \geq k_{b_i}$ 故 $b_{i+1} = (k_{a_i} \bmod k_{b_i}) \cdot c_i$

$\therefore rgcd(a_{i+1}, b_{i+1}) = rgcd(k_{b_i} c_i, (k_{a_i} \bmod k_{b_i}) c_i) = c_i = rgcd(a_i, b_i)$

从而证明 调用中的循环不变式 $\mathrm{rgcd}\left(a_i, b_i\right) = \mathrm{rgcd}\left(a_{i+1}, b_{i+1}\right)$

（四）证明 调用中的另一个不变式 $\gcd\left(a_k, b_k\right) = \mathrm{rgcd}\left(a_k, b_k\right)$（此处 $k \leq n$ k 可以等于 n）

① 首先 证明 $k = n$ 时候成立，由于 $b_n = 0$ 由拓展定义得 $\mathrm{rgcd}\left(a_n, b_n\right) = a_n = \gcd(a_n, b_n)$

② 由于 k = n 时 不具有普遍性，我们仍需在不使用 $\mathrm{rgcd}\left(a_n, b_n\right) = \gcd(a_n, b_n)$ 的情况下，

证明 $k = n - 1$ 时，成立。（注意，此处我们默认 $n \geq 2$）

由于 $b_n = 0$， 2.1 $a_{n-1} = b_{n-1} \cdot m \ (m \geq 1)$

$\therefore 0 = a_{n-1} \bmod b_{n-1}$ 则 $gcd(a_{n-1}, b_{n-1}) = gcd(b_{n-1}, 0) = b_{n-1}$

或 $0 = a_{n-1} < b_{n-1}$ $rgcd(a_{n-1}, b_{n-1}) = b_{n-1}$ $\therefore rgcd(a_{n-1}, b_{n-1}) = gcd(a_{n-1}, b_{n-1})$

2.2. $b_{n-1} = 1$

则 $gcd(a_{n-1}, b_{n-1}) = gcd(b_{n-1}, 0) = b_{n-1} = 1$

$rgcd(a_{n-1}, b_{n-1}) = 1$ $\therefore rgcd(a_{n-1}, b_{n-1}) = gcd(a_{n-1}, b_{n-1})$

2.3. $a_{n-1} = 0 < b_{n-1}$ 由（一）得 $n = 2$（若 $n > 2$ 则 $a_{n-1} > b_{n-1}$）

$rgcd(a_{n-1}, b_{n-1}) = b_{n-1}$ $gcd(a_{n-1}, b_{n-1}) = gcd(b_{n-1}, a_{n-1}) = b_{n-1}$

$\therefore rgcd(a_{n-1}, b_{n-1}) = gcd(a_{n-1}, b_{n-1})$

虽然 2.3 也用到了拓展定义，但 2.3 仅
在 $n=2$ 时出现，故而对 $n>2$ 的一般情况
2.3 情况不存在 即未使用拓展定义

故而 在 n > 2时，在未使用拓展定义情况下，我们证明了 $\mathrm{rgcd}\left(a_{n-1}, b_{n-1}\right) = \gcd(a_{n-1}, b_{n-1})$

故而 $\forall\, i \in N^* \cap i \in [1, n]$，有：

$$rgcd\left(a_i, b_i\right) = rgcd\left(a_{i+1}, b_{i+1}\right) = \cdots = rgcd\left(a_{n-1}, b_{n-1}\right)$$
$$= \gcd\left(a_{n-1}, b_{n-1}\right) = \cdots = \gcd(a_i, b_i)$$

（理论上，这里严谨的做法是使用数学归纳法）

取 $i = 1$；$rgcd\left(a_1, b_1\right) = \gcd\left(a_1, b_1\right)$；即 算法正确。

而对 $n=1$ 或 2 时特殊情况，在使用拓展定义后，已在上文证明

#

2 算法复杂度基础

2 算法复杂度基础

2.1 证明

试证明, $\forall \epsilon > 0. \lg n = O(n^{\epsilon})$. [1]

$$\lim_{n\to\infty}\left(\frac{\lg(n)}{n^{\epsilon}}\right)^{\frac{\infty}{\infty}} = \lim_{n\to\infty}\frac{\frac{1}{\ln(2)} * \frac{1}{n}}{n^{\epsilon-1}} = \lim_{n\to\infty}\frac{1}{\ln(2)\, n^{\epsilon}} \overset{\epsilon>0}{=} 0$$

$\therefore \lg n = O(n^{\epsilon})$

2.2 判断

若 $f(n) = O(n^2)$ 且 $g(n) = O(n)$, 判断以下结论是否正确. 如果不正确, 请举出一个反例:

1. $f(n) + g(n) = O(n^2)$. ✓
2. $f(n)/g(n) = O(n)$. × $f(n) = n^2 \quad g(n) = 1$
3. $g(n) = O(f(n))$. × $g(n) = n \quad f(n) = 1$
4. $f(n)g(n) = O(n^3)$. ✓

2.3 函数复杂度排序 (From Prof. Chaodong Zheng)

Sort the following functions from asymptotically smallest to asymptotically largest, indicating ties if there are any. You do not need to prove your answer. To simplify notation, write $f(n) \ll g(n)$ to denote $f(n) \in o(g(n))$ and $f(n) = g(n)$ to denote $f(n) \in \Theta(g(n))$ [2].

$n^{1/\lg n} > 1$

$\lg(\lg^* n)$	$2^{\lg^* n}$	$(\sqrt{2})^{\lg n}$ $\sqrt{n}$	n^2	$n!$	$(\lg n)!$
$(\frac{5}{4})^n$	n^3	$\lg^2 n$	$\lg(n!)$	2^{2^n}	$n^{1/\lg n} = n^{\log_n 2} = 2$
$\ln\ln n$	$\lg^* n$	$n \cdot 2^n$	$n^{\lg\lg n}$	$\ln n$	1
$2^{\lg n} = n$	$(\lg n)^{\lg n}$	e^n	$4^{\lg n}$ n^2	$(n+1)!$	$\sqrt{\lg n}$
$\lg^*(\lg(n))$	$2^{\sqrt{2\lg n}}$	n	2^n	$n \lg n$	$2^{2^{n+1}}$

$\lg n \gg \lg^* n$

$\lg n > \ln n$

$\lg^* n = t$

$2^{t-1} \leq 2 \;\geq n > 2^{t-2}$

$\lg t \ll t \ll 2^t$

$2^u \quad u^u \quad \lg^* u \quad 2^{\sqrt{2}\cdot\sqrt{u}} \quad 2^{\frac{1}{2}u} \quad 2^{2u} \quad u^2 \quad \sqrt{u} \quad u!$

$> 2^{\lg^* n} \gg \lg^* n \gg \lg(\lg^* n)$

$u^u \gg u! \gg 2^{2u} \gg 2^u \gg 2^{\frac{1}{2}u} \gg 2^{\sqrt{2}\sqrt{u}} \gg u^2 \gg \sqrt{u} > \lg^* u$

$\underbrace{(\lg n)^{(\lg n)}}_{n^{\lg\lg n}} \gg (\lg n)! \gg \underbrace{4^{\lg n}}_{n^2} \gg \underbrace{2^{\lg n}}_{n} \gg \underbrace{\sqrt{2}^{\lg n}}_{\sqrt{n}} \gg 2^{\sqrt{2\lg n}} \gg \lg^2 n \gg \ln n > \sqrt{\lg n} \gg \lg^*(\lg n) = \lg^* n - 1$

$\ln(\ln n)$

$n^{\lg\lg n}$ n^2 $\gg \ln n$ $= \lg^* n - 1$

$\ln(\ln n)$ $\gg \lg(\lg^* n)$

$2^{2^{n+1}} \gg 2^{2^n}$ $\gg \lg(\lg(\lg(n)))$

$\gg (n+1)! \gg n! \gg e^n \gg n\cdot 2^n \gg 2^n \gg \left(\frac{5}{4}\right)^n \gg n^3 \gg n^2 \gg n\lg n \gg \lg(n!) \gg n$

$\gg \lg^*(\lg(\lg(\lg(\lg n)))) + 4 \sim \lg^* n$

$(\lg n)^{(\lg n)} \sim n^{\lg\lg n}$ $(\lg n)^{(\lg n)} < 2^n$

$u^u < 2^{2^u}$ $\Rightarrow \lg\lg(\lg(n)) \gg 2^{\lg^* n}$

$\lg n = u \Rightarrow n = 2^u$ $u\lg u \ll 2^u$

$u^u = 2^{u\cdot\lg u}$

最终的顺序是： $2^{2^{n+1}} \gg 2^{2^n} \gg (n+1)! \gg n! \gg e^n \gg n\cdot 2^n \gg 2^n \gg \left(\frac{5}{4}\right)^n \gg (\lg n)^{\lg n} = n^{\lg\lg n} \gg (\lg n)! \gg n^3 \gg n^2 = 4^{\lg n} \gg n\cdot\lg n \gg \lg(n!) \gg n = 2^{\lg n} \gg \sqrt{2}^{\lg n} \gg 2^{\sqrt{2\lg n}} \gg \lg^2 n \gg \ln n \gg \sqrt{\lg n} \gg \ln\ln n \gg 2^{\lg * n} \gg \lg * n = \lg * (\lg n) \gg \lg(\lg * n) \gg n^{\frac{1}{\lg n}} = 1$