A1 算法基础

ddl: 2023/9/22 24:00

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)
```

2 算法复杂度基础

2.1 证明

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

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. $f(n)/g(n) = O(n)$.
3. $g(n) = O(f(n))$.
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].

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

---

1. 本次作业中 $\lg(n)$ 代表 $\log_2(n)$. ↩

2. $\lg^* n = \min\{i \geq 0 | \lg^{(i)} n \leq 1\}$. For better understanding of the function $\lg^* n$, you can refer to section 3.3 of CLRS (4th edition) for its definition. ↩