Week 1

软73 沈冠霖 2017013569

February 26, 2019

1 T1

证明 因为 $\exists c = 3, n_0 = 1$,对于$\forall n >= n_0$,有$cn^2 >= 2n$, 所以 2n 属于$O(n^2)$
因为$O(f(n)) + \theta(f(n)) = \theta(f(n))$，则原式成立

2 T2

证明 若f(n)=$\theta(g(n))$,则$\exists c > 0$,使得$n_0 > 0$，对于$\forall n > n_0, f(n) >= cg(n)$。与f(n)=o(g(n))矛盾，因此所有f(n)=$\theta(g(n))$构成的集合和所有f(n)=o(g(n)）构成的集合二者交集为空

3 T3

证明 取f(n) = sin(πn/2)+1,g(n) = 1,则
$\exists c = 2$,使得 $\exists n_0 = 1, \forall n > n_0, 0 <= f(n) <= cg(n)$,因此f(n)=O(g(n))
但是对于$\forall n_0 > 0$,总是$\exists n > n_0, f(n) = 0$,,不存在c¿0,使得$\exists n_0, \forall n > n_0, f(n) >= cg(n)$,因此$f(n) \neq \theta(g(n))$
同时$\exists c = 0.1, \forall n_0 > 0, \exists n > n_0$使得f(n) = 2,f(n)¿=cg(n),因此 $f(n) \neq o(g(n))$
因此$\exists f(n) = O(g(n)), f(n) \neq o(g(n))$且$f(n) \neq \theta(g(n))$,原式成立
但是这里举出的反例在自然数定义域中不单调递增，对于一个算法复杂度来说没有实际意义

4 T4

证明 不妨令n=某一正实数n_t时，f(n)¿=g(n),取c = 0.5,有$max(f(n_t), g(n_t)) >= c(f(n_t) + g(n_t))$，反之一样成立，则$\exists n_0 = 1, \forall n_t > n_0, max(f(n_t), g(n_t)) >= c(f(n_t) + g(n_t))$
而取c= 1，$\exists n_0 = 1, \forall n > 0, f(n)+g(n)-max(f(n), g(n)) = min(f(n), g(n)) >= 0$
综上，$max(f(n), g(n)) = \theta(f(n) + g(n))$

5 T5

第一问 **由增长率由最小到最大可划分如下等价类：（每行一个等价类）**
1,$n^{1/lgn}$
lg(lg*n)
lg*(lgn)
lg*(lgn)
lg*n
2^{lg*n}
ln(ln n)
$\sqrt{lgn}$
ln n
$(lgn)^2$
$(\sqrt{2})^{lgn}$
$n, 2^{lgn}$
$nlgn, lg(n!)$
$n^2, 4^{lgn}$
n^3
$2^{\sqrt{2lgn}}$
$(1.5)^n$
2^n
e^n
$n * 2^n$
$n^{lglgn}, (lgn)!, (lgn)^{lgn}$
$n!$
$(n+1)!$
2^{2^n}
$2^{2^{n+1}}$
方法：对于阶乘，幂取对数处理，对于对数，取指数处理。应用结论$lg(n!) = \theta(nlgn)$

第二问 $f(n) = 2^{2^{2n}} + (-1)^n 2^{2^{2n}}$，n为正整数