Lesson 02

时间复杂度

Time Complexity

案例

方法 1

找到所有 两个数 配对的可能

保留最高值

伪代码：

$Max2Sum(A, n)$

    $sum \leftarrow 0$
    for $i \leftarrow 1$ to n
        for $j \leftarrow 1$ to n
            if i! = j then
                if $A_i + i_j > sum$ then
                    $sum = A_i + A_j$

return sum

方法 2

循环一遍 找到最大值

再循环一遍 找到第二个大的值

伪代码：

$Max2Sum(A, n)$

    $max_1 \leftarrow 0$
    for $i \leftarrow 1$ to n
        if $A_i > max_1$ then
            $max_1 \leftarrow A_i$
    $max_2 \leftarrow 0$
    for $i \leftarrow 1$ to n
        if $A_i > max_2$ and A_i ! = max_1 then
            $max_2 \leftarrow A_i$

return $max_1 + max_2$

方法 3

跟方法 2 一样，但只走一趟

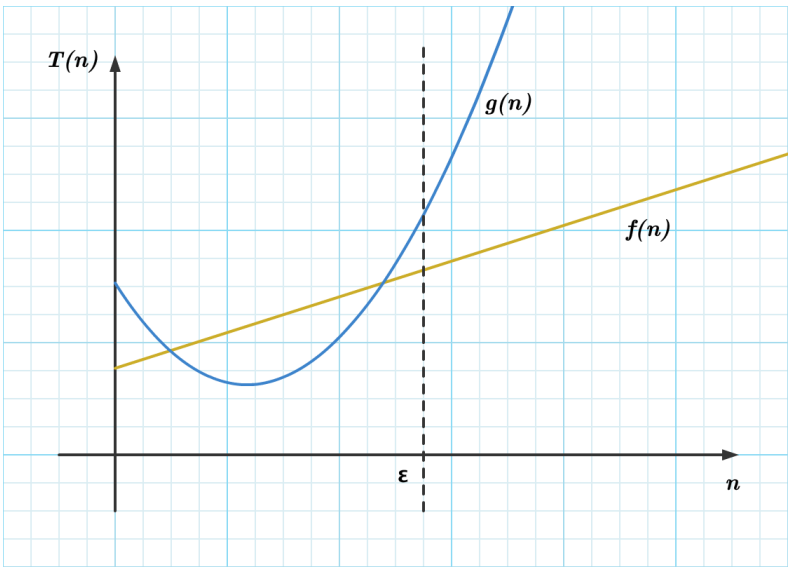
伪代码：

$Max2Sum(A, n)$
    $max_1 \leftarrow 0$
    $max_2 \leftarrow 0$
    for $i \leftarrow 1$ to n
        if $A_i > max_2$ then
            $max_2 \leftarrow A_i$
        if $A_i > max_1$ then
            $Swap(max_1, max_2)$
return $max_1 + max_2$

相对优势 与 绝对优势

大于 某个值之后

我们比较大于某个值 ϵ 后面的



相对优势

什么是：

通过对 时间公式 叠加 常量倍数 会影响比较结果

案例：

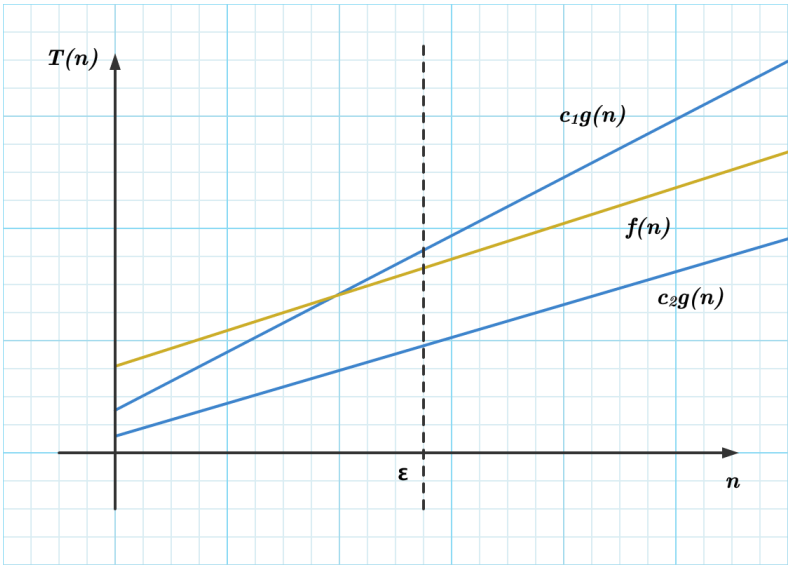
情况	$f(n) = n$	$g(n) = 2n, c \cdot g(n)$	更好
$n = 10, c = 1$	10	20	$f(n)$
$n = 100, c = 1$	100	200	$f(n)$
$n = 10, c = 1/10$	10	2	$g(n)$
$n = 100, c = 1/10$	100	20	$g(n)$

实际：

如果设备一样 那么一个比一个好

但是可以通过砸钱在硬件设备上，彻底改变结果

函数图：



$\exists\ \epsilon > 0$

when $n > \epsilon$ and $n \to \infty$

$\exists\ c_1 > 0, c_1 \cdot g(n) \geq f(n)$ and

$\exists\ c_2 > 0, f(n) \geq c_2 \cdot g(n)$

绝对优势

什么是：

通过对 时间公式 叠加 常量倍数 不会 影响比较结果

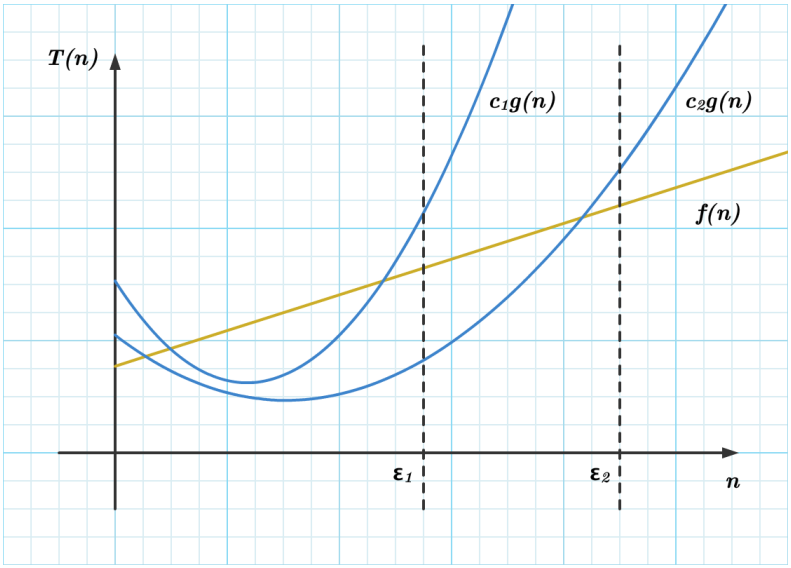
案例：

情况	$f(n)=n$	$g(n)=n^2, c\cdot g(n)$	更好
$n=2, c=1$	2	4	$f(n)$
$n=2, c=1/10$	2	0.4	$g(n)$
$n=20, c=1/10$	20	40	$f(n)$
$n=20, c=1/100$	20	4	$g(n)$
$n=200, c=1/100$	200	400	$f(n)$
$n=200, c=1/1000$	200	40	$g(n)$

实际：

就算通过砸钱在硬件设备上，当数据量大时也无法改变结果

函数图：



$\forall\ c > 0$
$\exists\ \epsilon > 0$
when $n > \epsilon$ and $n \to \infty$
$c \cdot g(n) \geq f(n)$ and

优势研究

侧重点：

应该优先专注 提高绝对优势

代码可读性：

当只有相对优势时，我们可以 专注代码可读性

速度档位

速度档位：

互相有相对优势的时间函数，被归类到一个速度档位内

n	时间增长慢	低档位	更好
n^2	时间增长快	高档位	不好

$\Theta(n)$

什么是：

如果 $T_1(n) = f(n)$ 和 $T_2(n) = g(n)$ 在一个速度档位上

就说 $f(n) = \Theta(g(n))$

含义：

意思就是 $f(n)$ 与 $g(n)$ 档位相同

集合含义：

$\Theta(g(n))$ 代表 跟 $g(n)$ 档位 相同的那些函数 所组成的集合

那么 $f(n) = \Theta(g(n))$

意思就是 $f(n) \in \Theta(g(n))$

$O(n)$

什么是：

如果 $T_1(n) = f(n)$ 和 $T_2(n) = g(n)$ 在一个速度档位上

或者 $T_1(n) = f(n)$ 比 $T_2(n) = g(n)$ 有绝对优势

就说 $f(n) = O(g(n))$

含义：

意思就是 $f(n)$ 档位 差不过 $g(n)$

集合含义：

$O(g(n))$ 代表 跟 $g(n)$ 档位 相同 以及 更好 的那些函数 所组成的集合

那么 $f(n) = O(g(n))$

意思就是 $f(n) \in O(g(n))$

练习

以下哪个是对的：

1. $n = \Theta(n)$
2. $n = \Theta(2n)$
3. $2n = \Theta(n)$

4. $n = \Theta(n^2)$
5. $n^2 = \Theta(n)$
6. $n = O(n)$
7. $n = O(2n)$
8. $2n = O(n)$
9. $n = O(n^2)$
10. $n^2 = O(n)$

相关推论 与 常用数学公式

证明档位

$\lim_{n\to\infty} \frac{f(n)}{g(n)}$

值	意义
1	档位一样
0	$f(n)$ 比 $g(n)$ 有绝对优势
∞	$g(n)$ 比 $f(n)$ 有绝对优势

多项式后面抹除

如果时间公式为一个多项式，只需要保留最高档位的部分

比如：

$T(n) = n^2 + 2n + 1 = O(n^2)$

常量系数抹除

如果时间公式为乘积，可以抹除常量系数

比如：

$T(n) = 3n^2 = O(n^2)$

log 相关公式

去底：

$\log_a n = O(\frac{\log_c n}{\log_c a}) = O(\log_c n) = O(\log n)$

去幂：

$\log n^a = O(a \log n) = O(\log n)$

去系数：

$\log_a bn = O(\log_a b + \log_a n) = O(\log_a n) = O(\log n)$

档位列表

$O(c) < O(\log n) < O(n) < O(n \log n) < O(n^2) < O(n^c) < O(c^n)$