

湖南大学-算法设计与分析课程



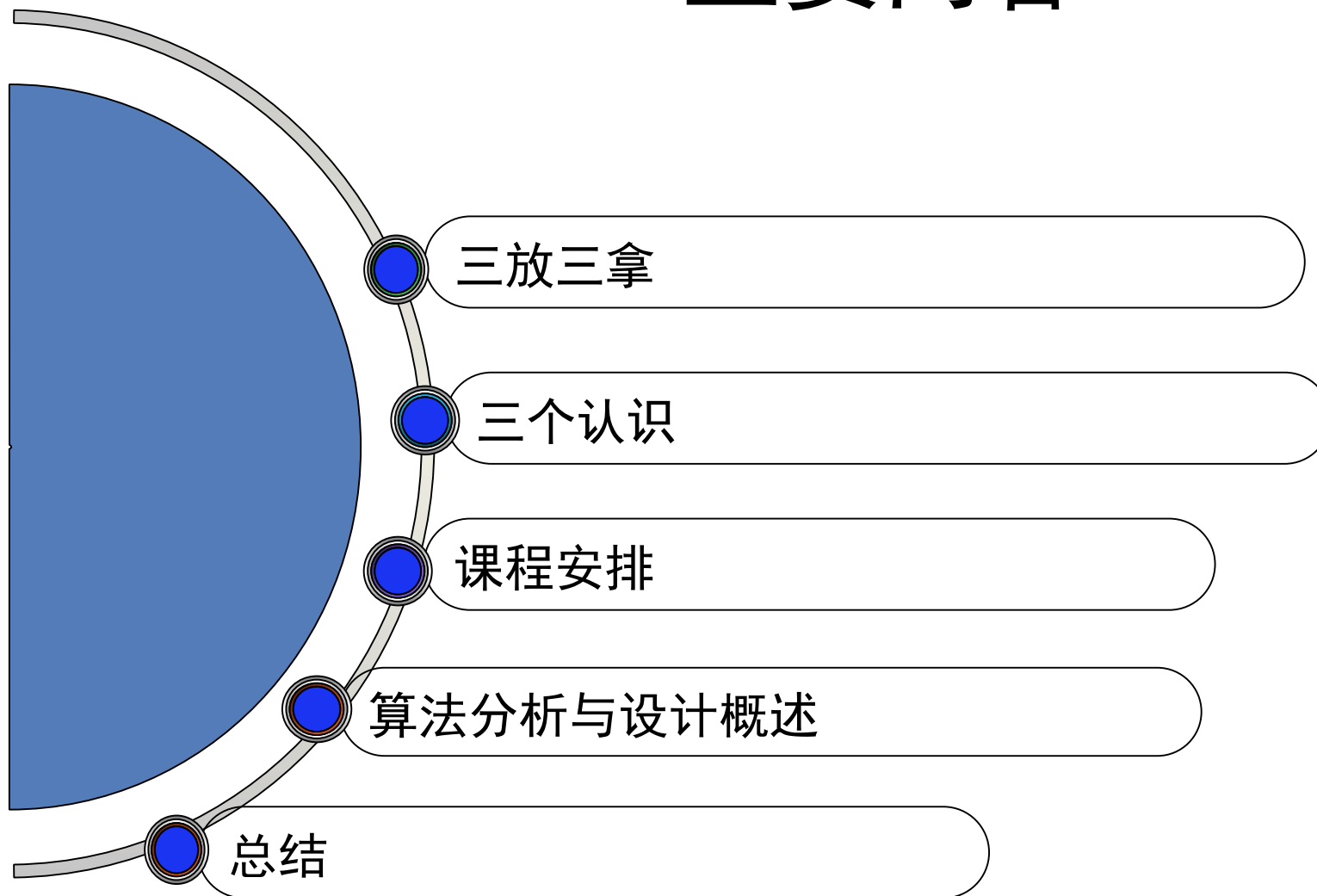
算法设计与分析-Lecture 1

姜文君

wenjj8a@163.com

办公室：信息科学与工程学院院楼326
2014-2015第二学期

主要内容



三 放



三拿

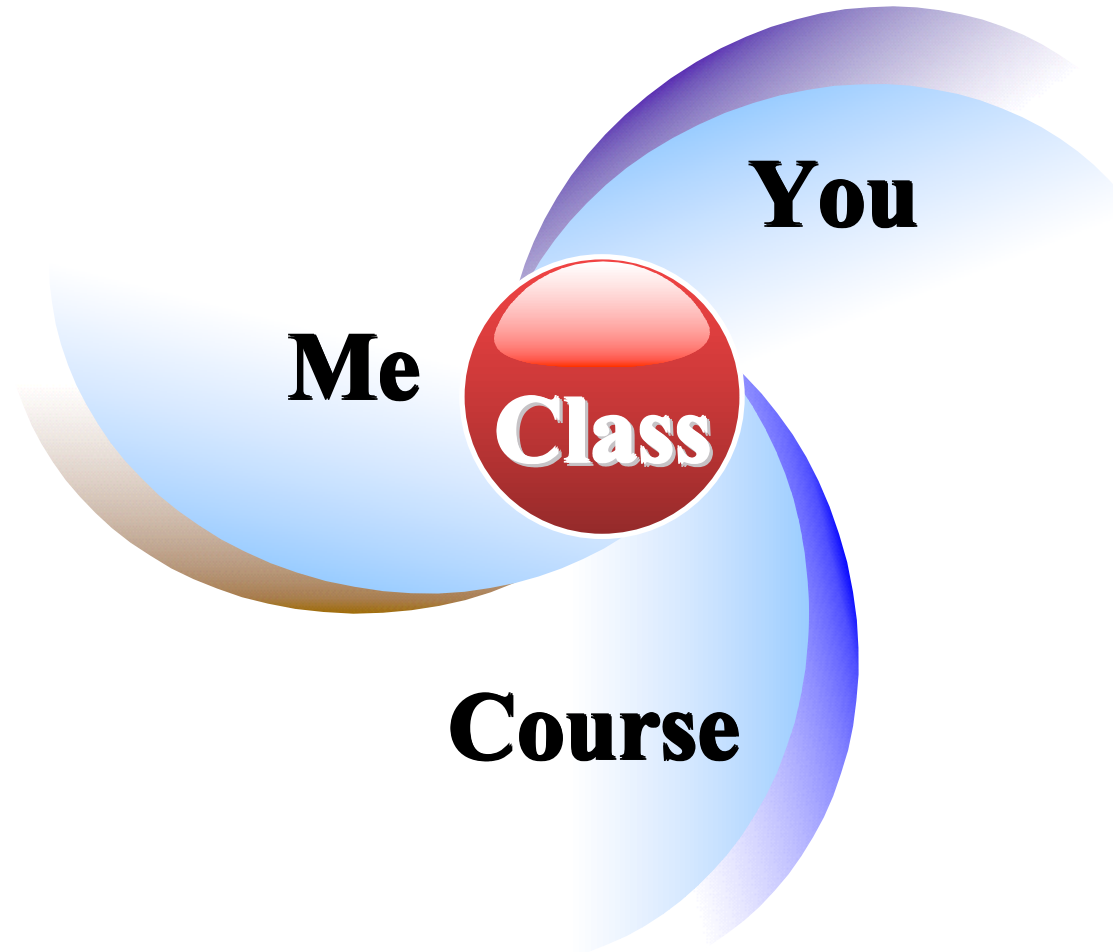


rlpis.com/395

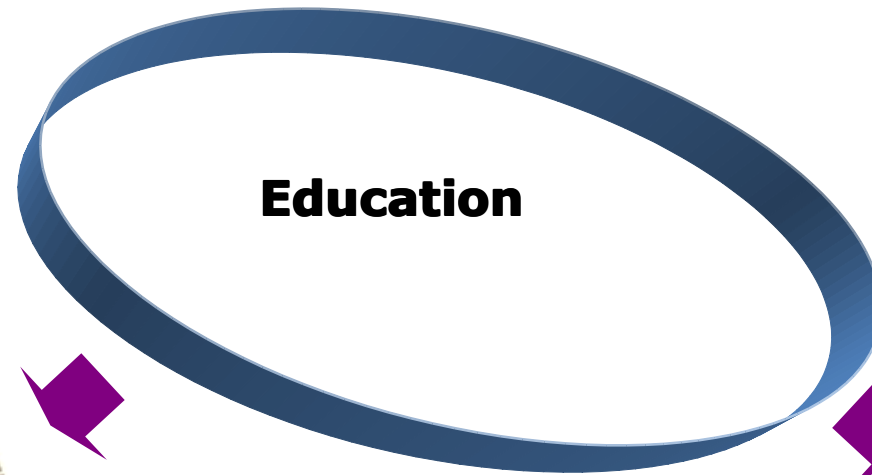
© iStockphoto.com



三个认识



Me



Education



HNU



HUST



CSU&TU



YOU

- 世界是你们的
- 课堂是你们的



分组讨论



课堂报告

Course



知识

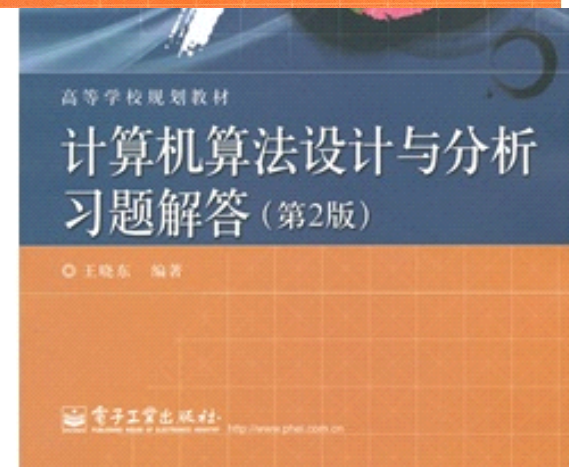
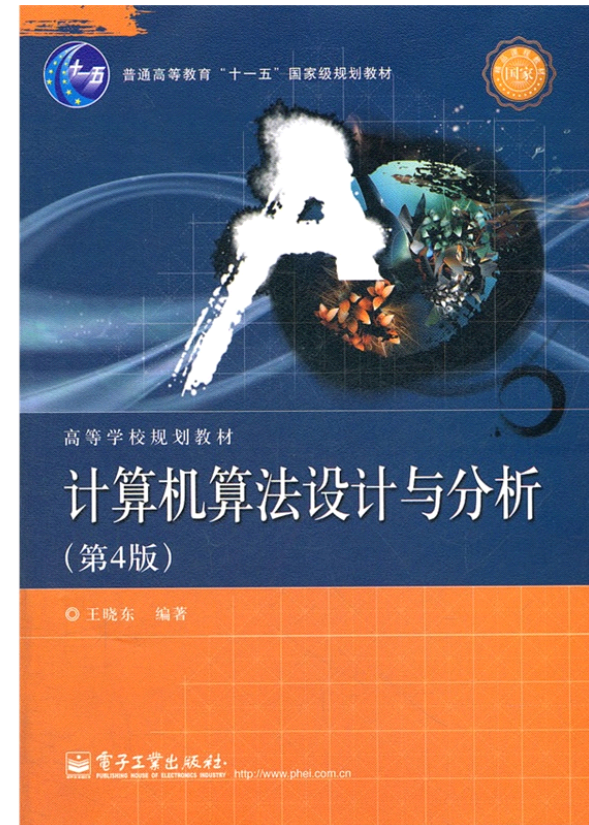


方法



关于课程

- 课程- 周二上午【3-4】
10:00-11:40, 中楼306
- 双周周四下午【7-8】
- 实验待定
- 教材、资料
 - 《计算机算法设计与分析（第4版）》王晓东 编著, 出版社: 电子工业出版社, 2012年.
 - 计算机算法设计与分析习题解答（第2版）
 - Wikipedia
 - Google



关于成绩

- 平时成绩 - 15%
 - 课堂参与、课后练习
- 实验成绩 - 25%
 - 实验准备、实验执行、实验总结报告
- 期中考试 - 20%
- 期末考试 - 40%

随机点名、三次未到者视为放弃本课程
三次迟到 == 一次未到

Welcome

算法(Algorithm)

- What is 算法?
- When do we need 算法?
- What 算法 is a good one?

与生活息息相关的算法



[新闻](#) [网页](#) [贴吧](#) [知道](#) [音乐](#) [图片](#) [视频](#) [地图](#) [百科](#) [文库](#) [更多>>](#)



[宝贝](#) [天猫](#) [店铺](#)

Q 宝宝坐车须谨慎 "安全座椅"平安出行

搜索

高级
搜索

毛呢外套 新款毛衣 [连衣裙](#) 时尚单鞋 进口零食 时尚保温杯 新款衬衫 新款童装 棉衣 时尚马丁靴 皮衣 时尚风衣

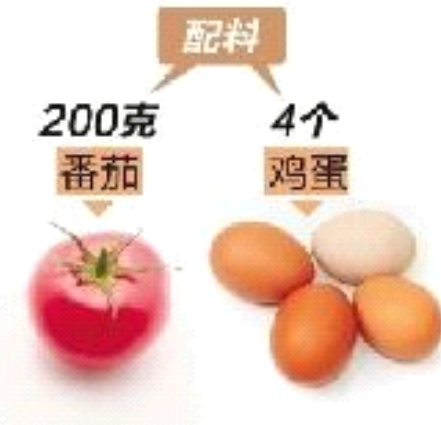
[更多](#)

What is 算法

解决问题的一种方法或一个过程。

- 算法是若干指令的有穷序列，满足性质：
 - (1)**输入**：有外部提供的量作为算法的输入。
 - (2)**输出**：算法产生至少一个量作为输出。
 - (3)**确定性**：组成算法的每条指令是清晰，无歧义的。
 - (4)**有限性**：算法中每条指令的执行次数是有限的，执行每条指令的时间也是有限的。

Example of 算法



程序(Program)

- 程序是算法用某种程序设计语言的具体实现。
- 程序可以不满足算法的性质(4)。
 - 例如操作系统，是一个在无限循环中执行的程序，因而不是一个算法。操作系统的各种任务可看成是单独的问题，每一个问题由操作系统中的一个子程序通过特定的算法来实现。该子程序得到输出结果后便终止。

算法 VS 程序

程序=算法+数据结构

主要贡献：

提出了结构化程序设计的思想
(逐步细化，或称为逐步求精，
stepwise refinement)



1984年图灵奖得主：
瑞士的Niklaus Wirth

算法 VS 程序

“他引导了并将继续引导这个星球上的程序员”

主要贡献：

1950年代ALGOL语言的一个主要贡献者。ALGOL语言结构清晰，数学基础严谨。

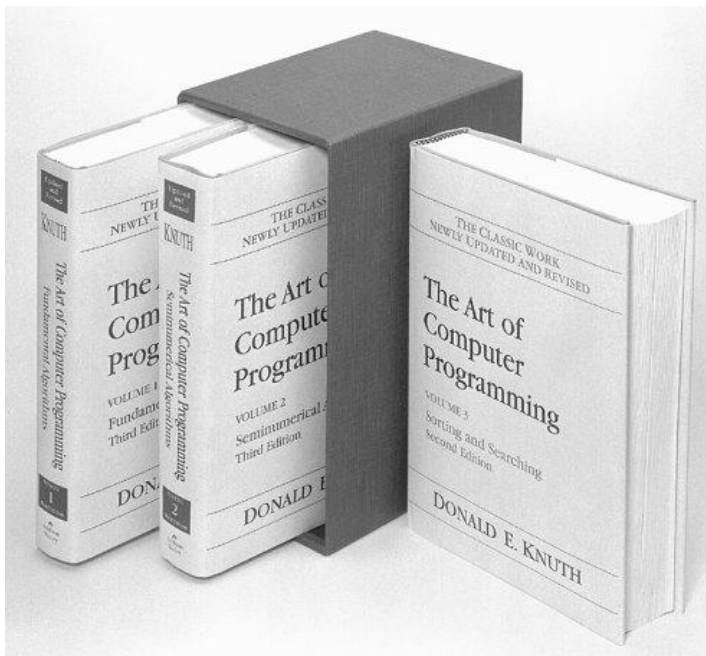
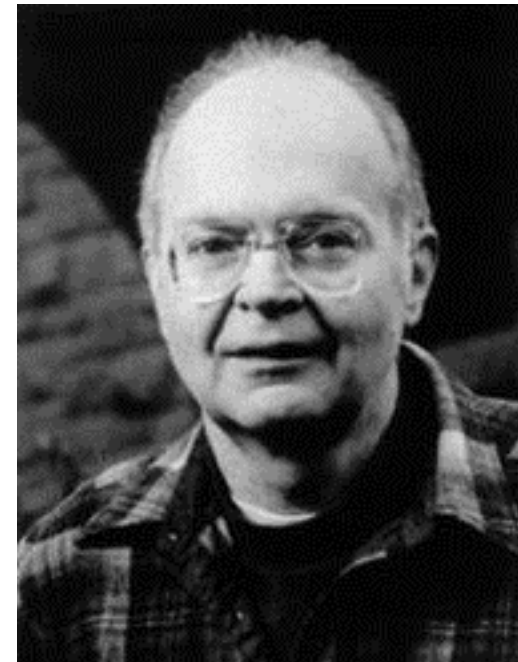


1972年图灵奖得主：荷兰的
Edsger Wybe Dijkstra

迪克斯特拉 04/01/1930-08/06/2002

算法 VS 程序

在算法分析和程序语言设计领域的杰出贡献，特别是其著名的“Art of Computer Programming”系列丛书。



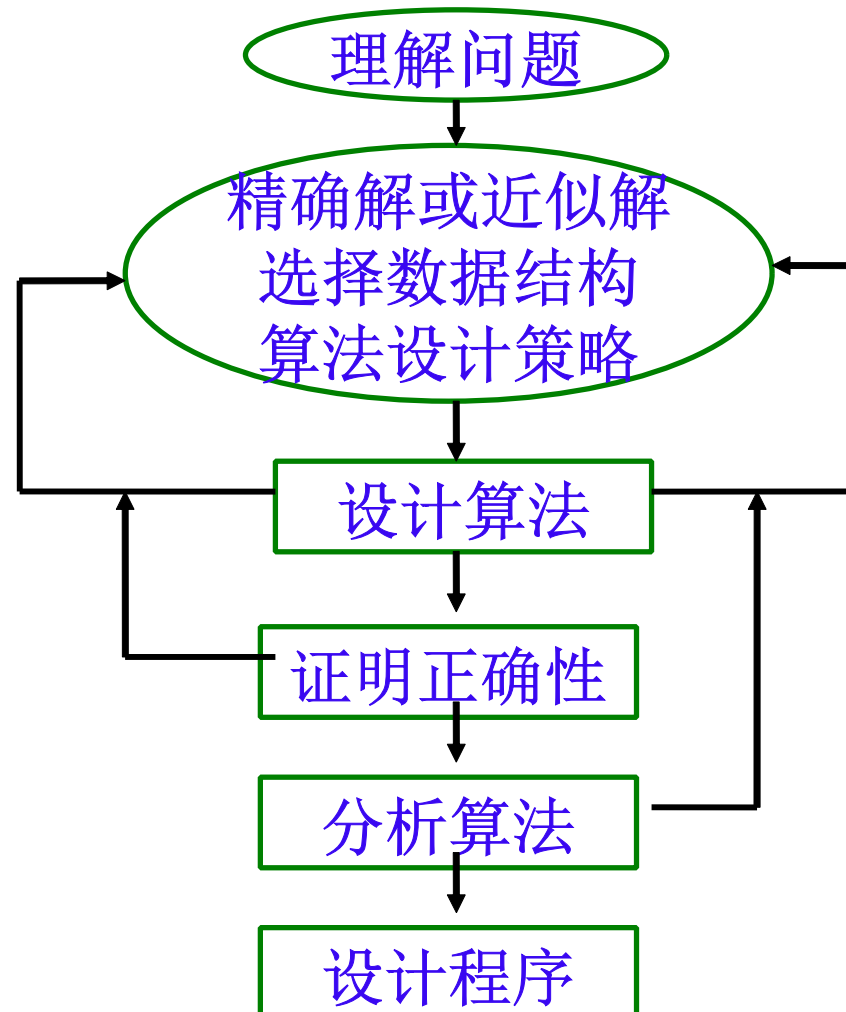
1974年图灵奖得主：美国的 Donald E. Knuth（高德纳）

When do we need 算法



Everything, Everytime, Everywhere

问题求解(Problem Solving)



算法: A Good One or Not

- 算法分析
- 质量
 - 正确性、可靠性、健壮性、可读性
- 效率
 - 时间复杂度、空间复杂度

提高算法质量

请看两组操作：

(1) $a := a + b;$ $b := a - b;$ $a := a - b;$

(2) $t := a;$ $a := b;$ $b := t;$

功能：“交换变量a、b中的数据”

第一组操作节省了一个存储空间，但失去了可读性。

拒绝偷工减料，拒绝豆腐渣工程！

算法复杂性分析

-时间和空间

- 算法复杂性 = 算法所需要的计算机资源
- 算法的时间复杂性 $T(n)$;
- 算法的空间复杂性 $S(n)$ 。
 - n 是问题的规模（输入大小）。

算法的时间复杂性

- (1) **最坏情况**下的时间复杂性
- $T_{\max}(n) = \max \{ T(I) \mid \text{size}(I)=n \}$
- (2) **最好情况**下的时间复杂性
- $T_{\min}(n) = \min \{ T(I) \mid \text{size}(I)=n \}$
- (3) **平均情况**下的时间复杂性
- $$T_{\text{avg}}(n) = \sum_{\text{size}(I)=n} p(I)T(I)$$
- 其中I是问题的规模为n的实例， $p(I)$ 是实例I出现的概率。

算法渐近复杂性

- $T(n) \rightarrow \infty$, as $n \rightarrow \infty$;
- $(T(n) - t(n)) / T(n) \rightarrow 0$, as $n \rightarrow \infty$;
- $t(n)$ 是 $T(n)$ 的渐近性态，为算法的渐近复杂性。
 - 在数学上， $t(n)$ 是 $T(n)$ 的渐近表达式，是 $T(n)$ 略去低阶项留下的主项。它比 $T(n)$ 简单。

百万富翁其实可能有108万、170万..

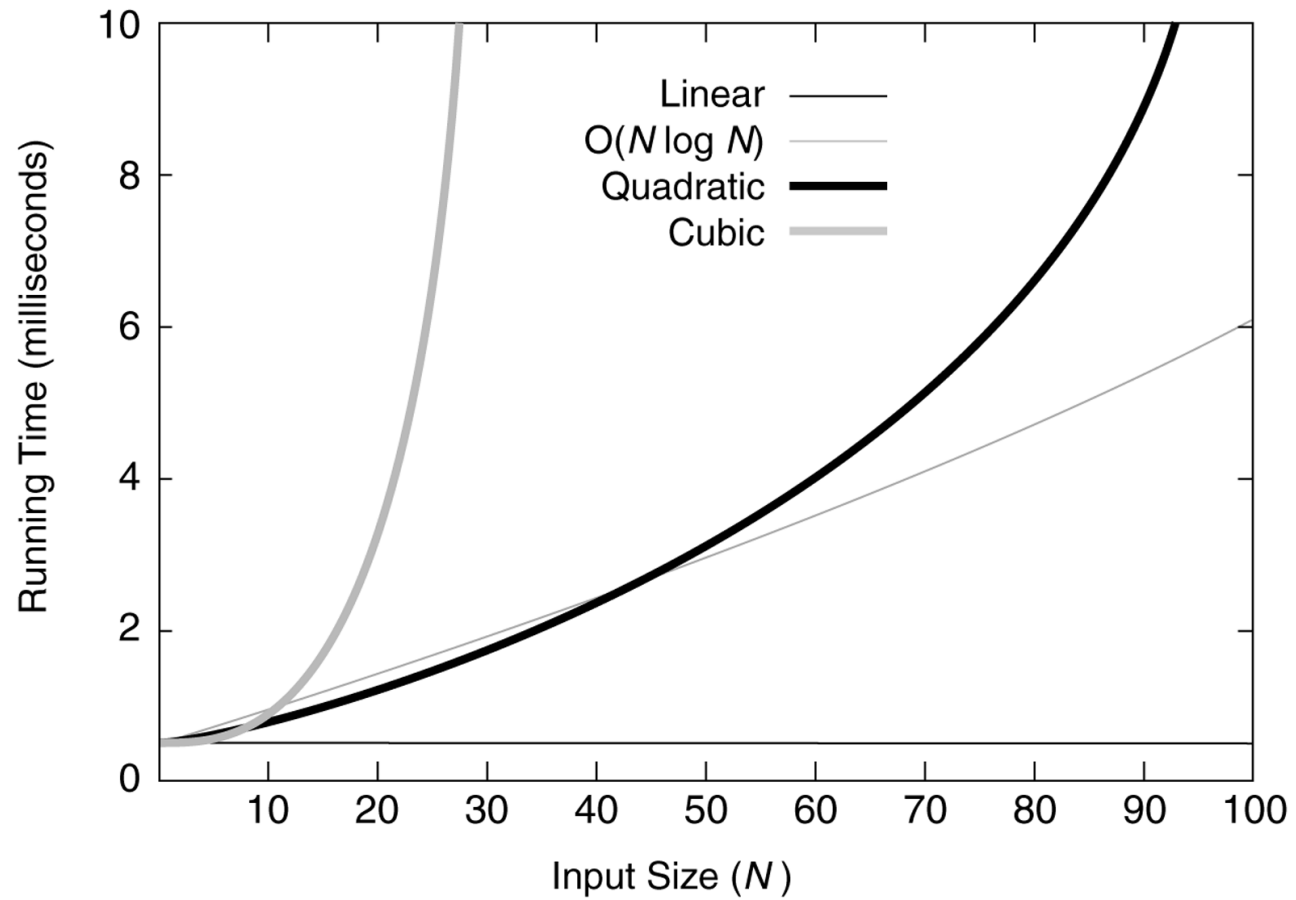
渐近分析的记号

- 在下面的讨论中，对所有 n ， $f(n) \geq 0$ ， $g(n) \geq 0$ 。
- (1) **渐近上界记号** O
- $O(g(n)) = \{f(n) \mid \text{存在正常数 } c \text{ 和 } n_0 \text{ 使得对所有 } n \geq n_0 \text{ 有: } 0 \leq f(n) \leq cg(n)\}$
- (2) **渐近下界记号** Ω
- $\Omega(g(n)) = \{f(n) \mid \text{存在正常数 } c \text{ 和 } n_0 \text{ 使得对所有 } n \geq n_0 \text{ 有: } 0 \leq cg(n) \leq f(n)\}$

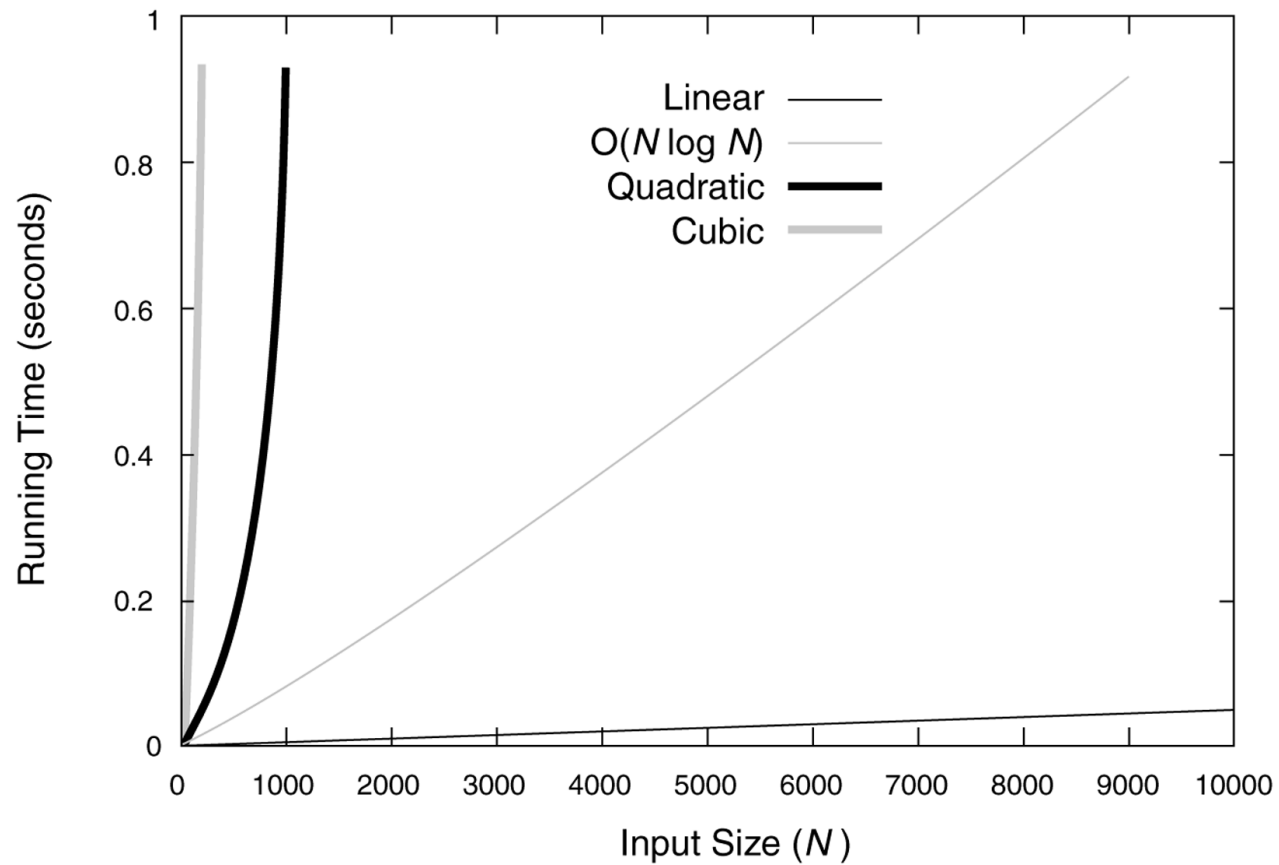
算法分析中常见的复杂性函数

FUNCTION	NAME
c	Constant
$\log N$	Logarithmic
$\log^2 N$	Log-squared
N	Linear
$N \log N$	$N \log N$
N^2	Quadratic
N^3	Cubic
2^N	Exponential

小规模数据



中等规模数据



算法分析方法

- 例：顺序搜索算法

```
template<class Type>
int seqSearch(Type *a, int n, Type k)
{
    for(int i=0;i<n;i++)
        if (a[i]==k) return i;
    return -1;
}
```

- (1) $T_{\max}(n) = \max \{ T(I) \mid \text{size}(I)=n \} = O(n)$
- (2) $T_{\min}(n) = \min \{ T(I) \mid \text{size}(I)=n \} = O(1)$
- (3) 在平均情况下，假设：
 - (a) 搜索成功的概率为 p ($0 \leq p \leq 1$);
 - (b) 在数组的每个位置 i ($0 \leq i < n$) 搜索成功的概率相同，均为 p/n 。

$$\begin{aligned}
 T_{avg}(n) &= \sum_{\text{size}(I)=n} p(I)T(I) \\
 &= \left(1 \cdot \frac{p}{n} + 2 \cdot \frac{p}{n} + 3 \cdot \frac{p}{n} + \dots + n \cdot \frac{p}{n} \right) + n \cdot (1 - p) \\
 &= \frac{p}{n} \sum_{i=1}^n i + n(1 - p) = \frac{p(n+1)}{2} + n(1 - p)
 \end{aligned}$$

算法分析的基本法则

非递归算法

- (1) for / while 循环
- 循环体内计算时间*循环次数；
- (2) 嵌套循环
- 循环体内计算时间*所有循环次数；
- (3) 顺序语句
- 各语句计算时间相加；
- (4) if-else语句
- if语句计算时间和else语句计算时间的较大者。

```

template<class Type>
void insertion_sort(Type *a, int n)
{
    Type key; // cost times
    for (int i = 1; i < n; i++){ // c1 n
        key=a[i]; // c2 n-1
        int j=i-1; // c3 n-1
        while( j>=0 && a[j]>key ){ // c4 sum of ti
            a[j+1]=a[j]; // c5 sum of (ti-1)
            j--; // c6 sum og (ti-1)
        }
        a[j+1]=key; // c7 n-1
    }
}

```

$$T(n) = c_1 n + c_2(n-1) + c_3(n-1) + c_4 \sum_{i=1}^{n-1} t_i + c_5 \sum_{i=1}^{n-1} (t_i - 1) + c_6 \sum_{i=1}^{n-1} (t_i - 1) + c_7(n-1)$$

- 在最好情况下, $t_i=1$, for $1 \leq i < n$;

$$T_{\min}(n) = c_1 n + c_2(n-1) + c_3(n-1) + c_4(n-1) + c_7(n-1)$$

$$= (c_1 + c_2 + c_3 + c_4 + c_7)n - (c_2 + c_3 + c_4 + c_7) = O(n)$$

$$\sum_{i=1}^{n-1} (i+1) = \frac{n(n+1)}{2} - 1 \quad \sum_{i=1}^{n-1} i = \frac{n(n-1)}{2}$$

- 在最坏情况下, $t_i \leq i+1$, for $1 \leq i < n$;

$$T_{\max}(n) \leq c_1 n + c_2(n-1) + c_3(n-1) +$$

$$c_4 \left(\frac{n(n+1)}{2} - 1 \right) + c_5 \left(\frac{n(n-1)}{2} \right) + c_6 \left(\frac{n(n-1)}{2} \right) + c_7(n-1)$$

$$= \frac{c_4 + c_5 + c_6}{2} n^2 + \left(c_1 + c_2 + c_3 + \frac{c_4 - c_5 - c_6}{2} + c_7 \right) n - (c_2 + c_3 + c_4 + c_7)$$

- 对于输入数据 $a[i]=n-i, i=0,1,\dots,n-1$ ，算法insertion_sort 达到其最坏情形。因此，

$$T_{\max}(n) \geq \frac{c_4 + c_5 + c_6}{2} n^2 + \left(c_1 + c_2 + c_3 + \frac{c_4 - c_5 - c_6}{2} + c_7 \right) n - (c_2 + c_3 + c_4 + c_7)$$
$$= \Omega(n^2)$$

- 由此可见， $T_{\max}(n) = \Theta(n^2)$

最优算法

- 问题的计算时间下界为 $\Omega(f(n))$ ，则计算时间复杂性为 $O(f(n))$ 的算法是最优算法。
- 例如，排序问题的计算时间下界为 $\Omega(n \log n)$ ，计算时间复杂性为 $O(n \log n)$ 的排序算法是最优算法。堆排序算法是最优算法。

总结

三放三拿来上课，
知识、方法双收获。

算法具有四性质：

输入、输出、确定、有限。

算法分析话时空，

大O大 Ω 最常用。

勤学勤练勤思考，

登高自卑千古理！

谢谢!

E-mail: wenjj8a@163.com

湖南大学-算法分析与设计