程序设计实习（实验班-2023春）
课程介绍

授课教师：姜少峰

助教：张宇博 楼家宁

Email: shaofeng.jiang@pku.edu.cn

个人履历

- 现任北京大学计算机学院 前沿计算研究中心 助理教授，博雅青年学者
- 香港大学博士->以色列魏茨曼科学院博士后->芬兰阿尔托大学助理教授->北大

我的研究

使用数学方法严格证明算法性能，不做实验

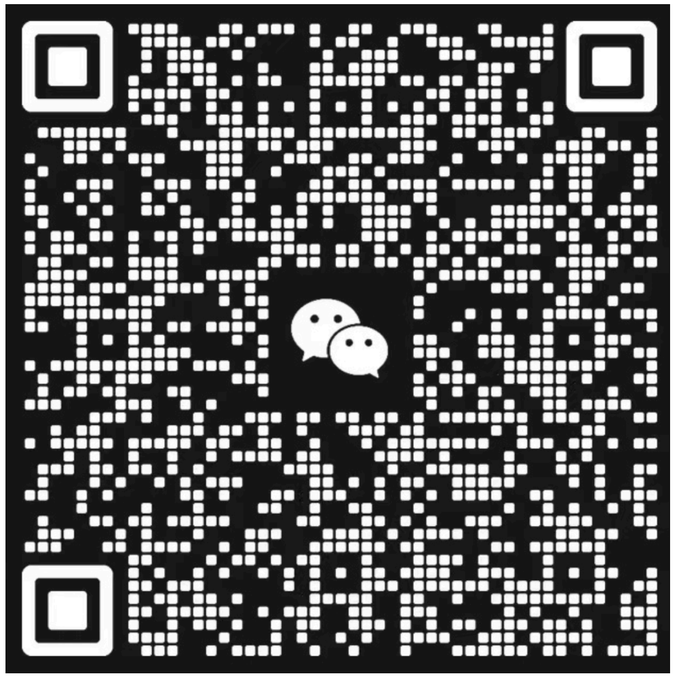
- 理论计算机科学 (Theoretical computer science)，侧重算法研究
  - 数据科学基础Foundations of Data Science
  - 大数据算法Algorithms for Massive Datasets
- 我的相关编程经验 *
  - 本科期间ACM-ICPC regional金奖，博士期间担任香港大学ACM-ICPC教练
  - 我的研究也经常考虑实际，实验验证一些大数据算法

课程资料

- 课程讲义和其他阅读材料
  - 教学网
  - 课程网页：https://shaofengjiang.cn/programming-course/
- 作业
  - 多数作业为编程作业，在http://cssyb.openjudge.cn在线提交和实时评测
  - 少数为实验报告，请提交到教学网（喜欢手写的同学也请拍照提交）

第一次使用需要把自己加入小组；
请把自己的小组昵称改成学号

答疑与联系方式

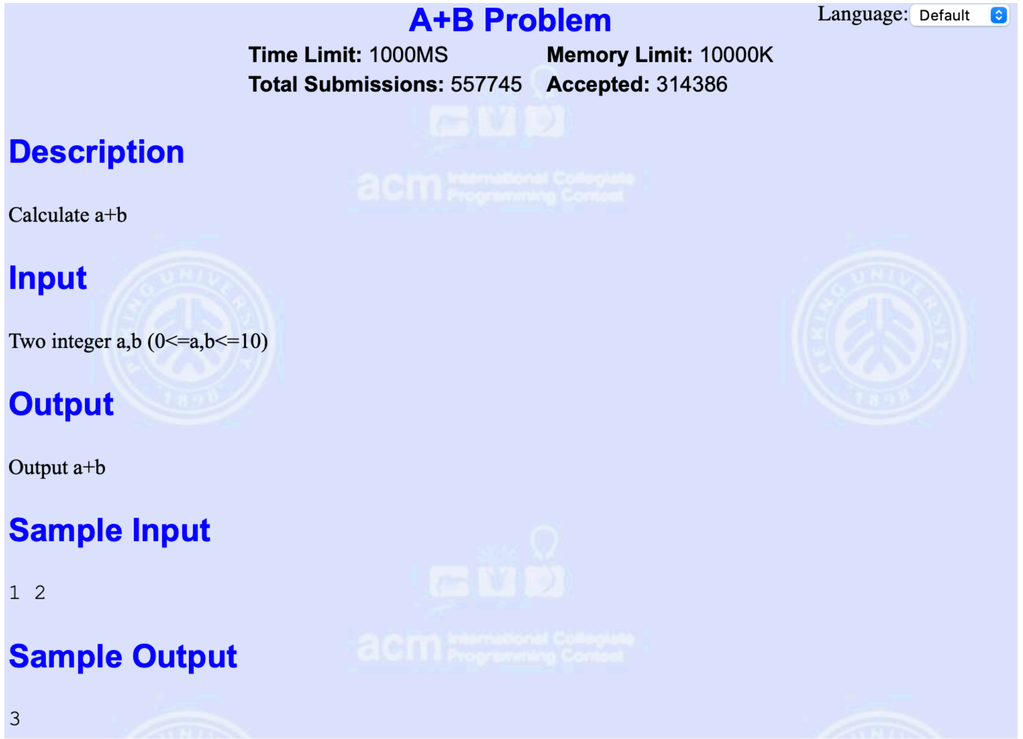


- 助教
  - 张宇博：北大图灵班本科，我课题组博士一年级研究生
  - 楼家宁：北大图灵班本科大四学生，将要在我课题组开始博士研究
- 答疑：Email、微信群、上机课（无考勤、无额外内容，主要是答疑）
  - shaofeng.jiang@pku.edu.cn，zhangyubo18@pku.edu.cn, loujn@pku.edu.cn
  - 上机课时间：周六下午1:00 - 3:00 ，理科一号楼1235（46-100号机器）

课程内容概述

程序设计

“程序设计”是为了解决“问题”



A+B Problem

Language: Default

Time Limit: 1000MS **Memory Limit:** 10000K
Total Submissions: 557745 **Accepted:** 314386

Description

Calculate a+b

Input

Two integer a,b (0<=a,b<=10)

Output

Output a+b

Sample Input

```
1 2
```

Sample Output

```
3
```

第一部分：程序设计的科学角度

- 问题已经清晰建模为“计算问题”
- 挑战：如何设计高效算法
- 有了算法又如何实现?

算法理论研究的核心

尤其关注实际效果，很多挑战无法被算法理论涵盖

算法：“传统”与“现代”

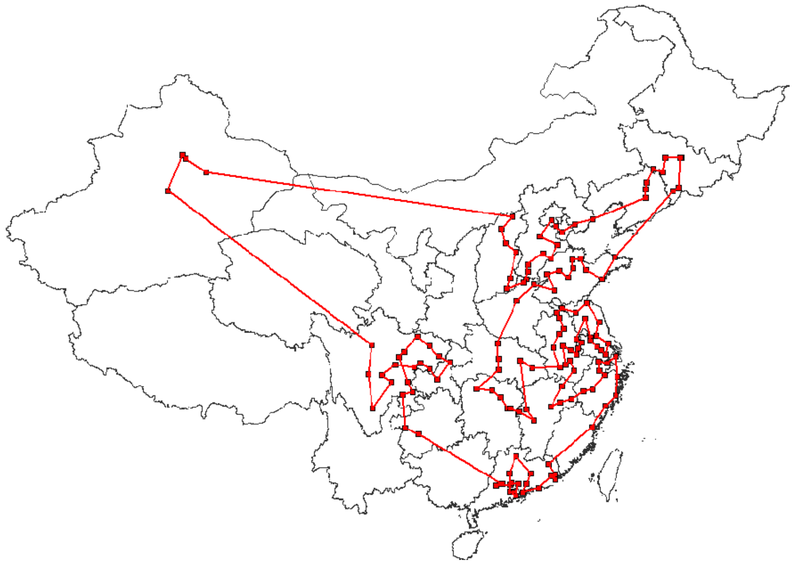
- 传统算法设计：
  - 有效算法 = 多项式时间
  - 追求“精确解”—尤其是“经典”算法/数据结构
  - 例子：传统的“算法与数据结构”课程的主要内容，如排序/搜索/字典数据结构；图上的最小生成树/拓扑排序/最短路

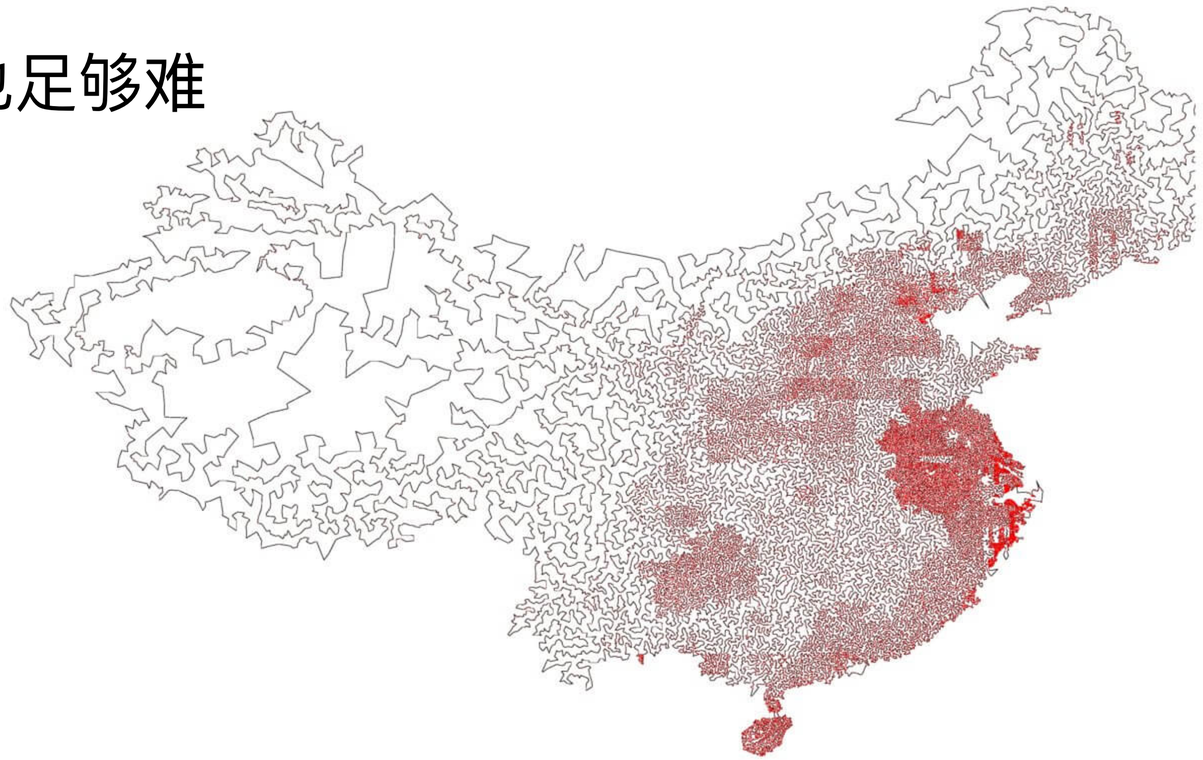
NP-hardness

暂可以粗略理解为“无多项式时间算法”

实际中的问题通常更为复杂，NP-hard更加常见

- 但是很多问题，即便是“简单”的问题，都可能是NP-hard
- TSP是一个典型：足够简单，但也足够难





近似算法

- “近似算法”研究领域：探索NP-hard问题在多项式时间内可以近似到多么精确
- 另外，即使是本来存在多项式时间算法的问题，允许近似解可潜在改进复杂度

如何衡量近似算法：近似比

最大化问题需要对应修改

- 对于优化（最小化）问题：

  最小化问题的近似比在$[1,\infty)$范围，

  - 一般采用近似比衡量解的质量：最坏情况下的$\frac{\mathrm{ALG}}{\mathrm{OPT}}$（这是一种相对误差）
  - 形式化：$\max_{I} \frac{\mathrm{ALG}(I)}{\mathrm{OPT}(I)}$
  - 称ALG是α-近似的，如果对任何input I，都有$\mathrm{ALG}(I) \leq \alpha \cdot \mathrm{OPT}(I)$
- 这个比值通常是通过分析论证得到

一般无法从数据验证，因为I可以有无穷种取法

其他误差衡量?

- 绝对误差：|ALG - OPT| $\leq \epsilon$
  - 当OPT较小时更弱：极端情况OPT = 0
  - 对于具有下界的问题更适用（比如对人类体验来说，精度小于1毫米没意义）
- 两种的折衷：$\mathrm{ALG} \leq \alpha \mathrm{OPT} + \beta$
- 但相对误差是研究的主流

考虑近似解对解决问题是否有意义?

- 实际应用中，通常“没有必要”追求精确解：
  - 一个实际问题是需要建模成计算问题的，这一步建模本身就是一种近似
  - 建模后，数据的采集也不是精确的过程，常混有一定噪音
  - 因此，算法即使得到精确解也依然是某种近似
- 另外，算法设计时考虑的是最坏情况的近似比，但在实际数据上经常表现更好

举例：TSP问题的发展

- 近似到常数也是NP-hard
- 2-近似：最小生成树（MST）
- 1.5-近似（Christofides 1976)：先MST再在奇数度顶点上做最小权匹配
- $1.5 - 10^{-36}$-近似 (Karlin, Klein, Gharan 2021) 91 pages…

举例：线性时间近似平面点集直径

- 输入：n个二维平面上的点$x_1, \ldots, x_n$
- 输出：对于直径$\max_{i,j} \|x_i - x_j\|_2$的估计

精确解需要$O(n^2)$时间

- 2-近似算法：选任一个点作为起点，例如x_1，返回x_1到距离x_1最远点的距离

随机性

- 类似地，另一种意义上的“近似”是引入“随机性”
- 随机算法：算法自身有随机性，即使对于确定性的输入也得到随机的输出
- 随机算法的性能保证：
  - 蒙特卡洛：有一定概率出错，不出错时有很好的性能保证
  - 拉斯维加斯：一定不出错，但是运行时间等资源消耗可以是随机的
- 相对确定性算法：由于保证变弱，通常换来大幅度的性能提升

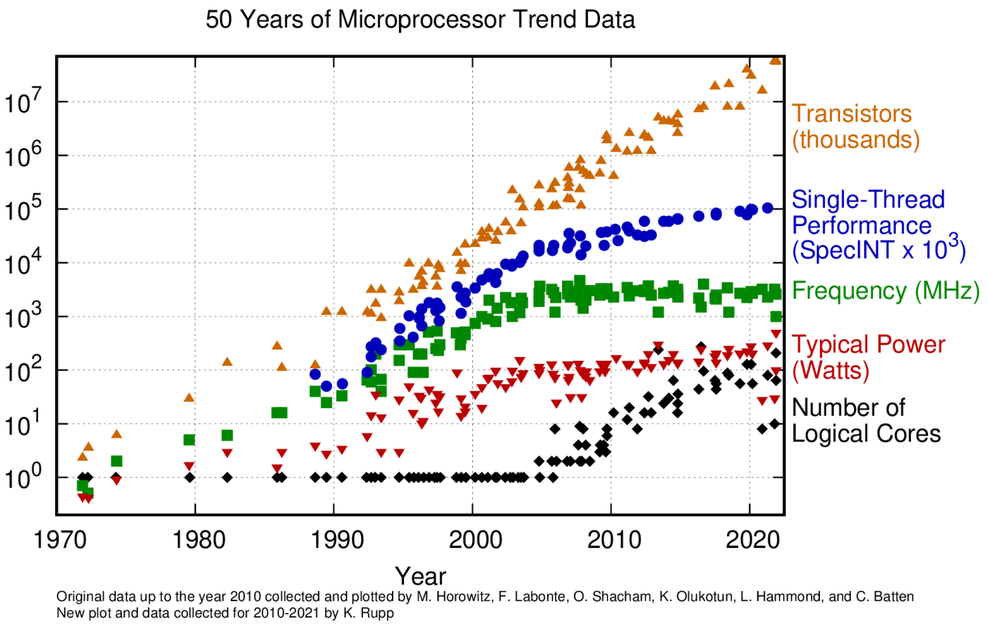
近似 + 随机

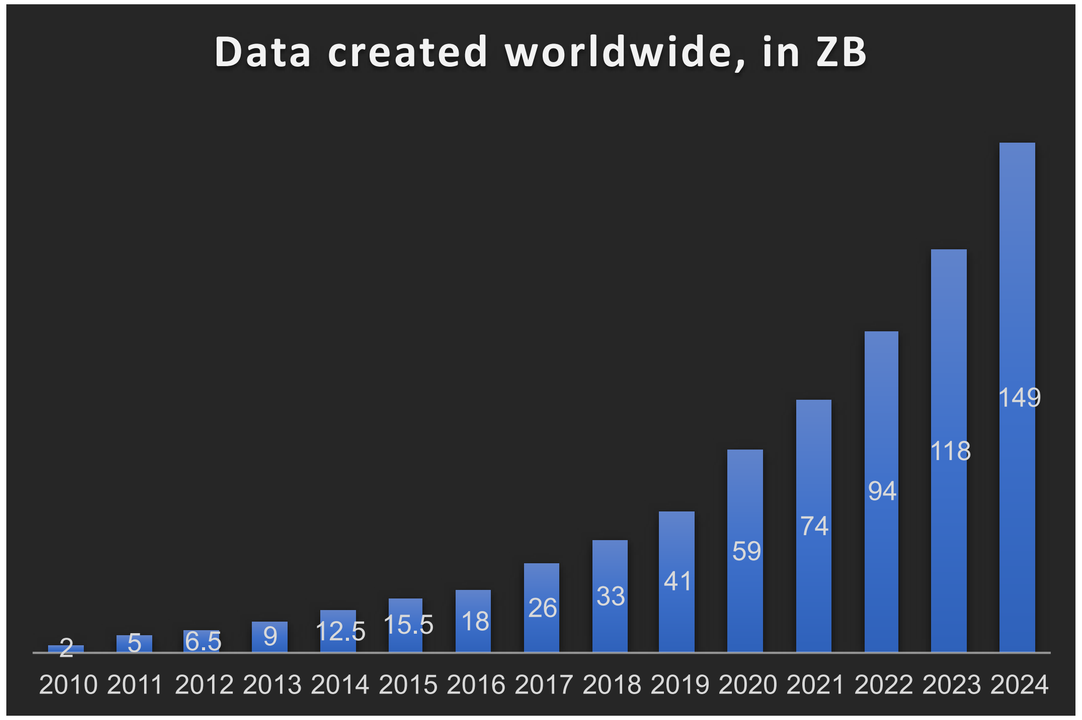
是现代算法设计的大趋势

大数据的时代

甚至线性时间都不够高效

有效算法 = 多项式时间算法?



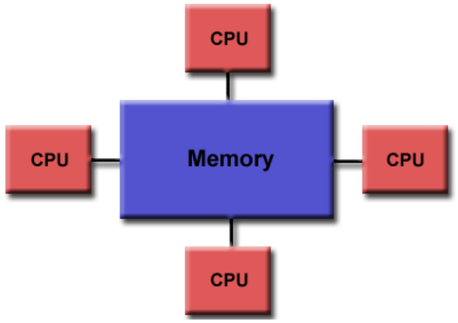


单个CPU：每秒10^{10}浮点运算；超级计算机：每秒10^{18}浮点运算

即使仅扫描一遍数据也不切实际；单CPU性能增速跟不上数据

“亚线性”计算模型

- 数据流算法：扫一遍数据流，用较小的空间计算
  - 路由器/物联网设备上直接做数据分析
- 并行算法

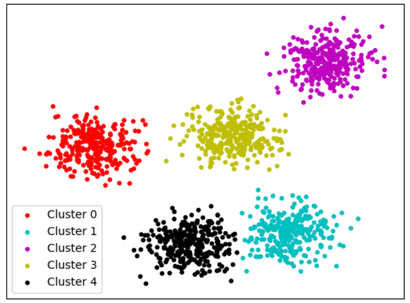


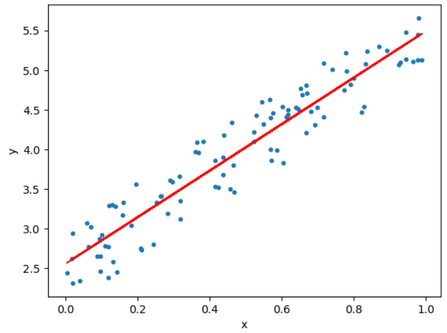
MapReduce框架是分布式计算最流行的软件实现

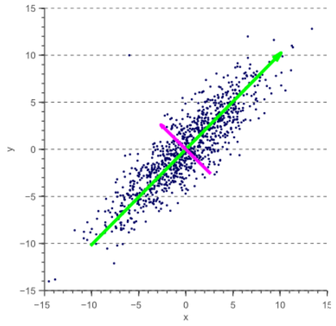
- 亚线性时间算法
  - 如何不遍历整个数据库，而是使用较少查询来进行计算?

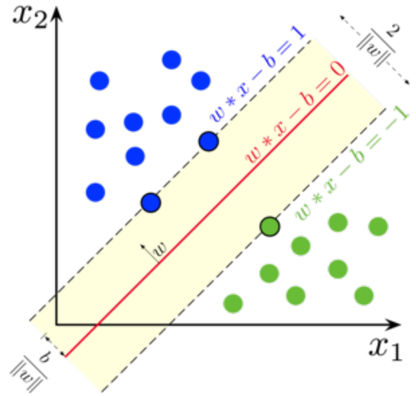
数据科学

- 脱胎于统计，使用计算的方法来“理解”大数据的一些总体性质
- 聚类，回归，主成分分析，分类...









- 这些问题的算法，尤其是大数据下的算法，成为了新的研究焦点

Introduction, *Foundations of Data Science*

Blum, Hopcroft and Kannan

- Computer science … began in the 1960's.
- In the 1970's, algorithms were added as an important component… The emphasis was on making computers useful.
- Today, fundamental change takes place … the focus is more on a wealth of applications.

> The enhanced ability to observe, collect, and store data in the natural sciences, in commerce, and in other fields calls for a change in our understanding of data and how to handle it in the modern setting. The emergence of the web and social networks as central aspects of daily life presents both opportunities and challenges.

- … this book covers the theory we expect to be useful in the next 40 years, just as an understanding of automata theory, algorithms, and related topics gave students an advantage in the last 40 years.

大数据 + 数据科学

是现代算法设计的新焦点

第一部分大纲：现代算法初探

- 随机算法及其实现
- 哈希方法及其在大数据上的应用
- 常见相似度/距离度量及其有效算法
- 几何近似算法
- 降维
- 压缩感知 & 数学优化

- 亚线性时间算法/并行算法
- 其他：聚类，社交网络

经典算法设计方法

- 递归方法
- 贪心方法等

部分内容基于Stanford CS 168: The Modern Algorithmic Toolbox

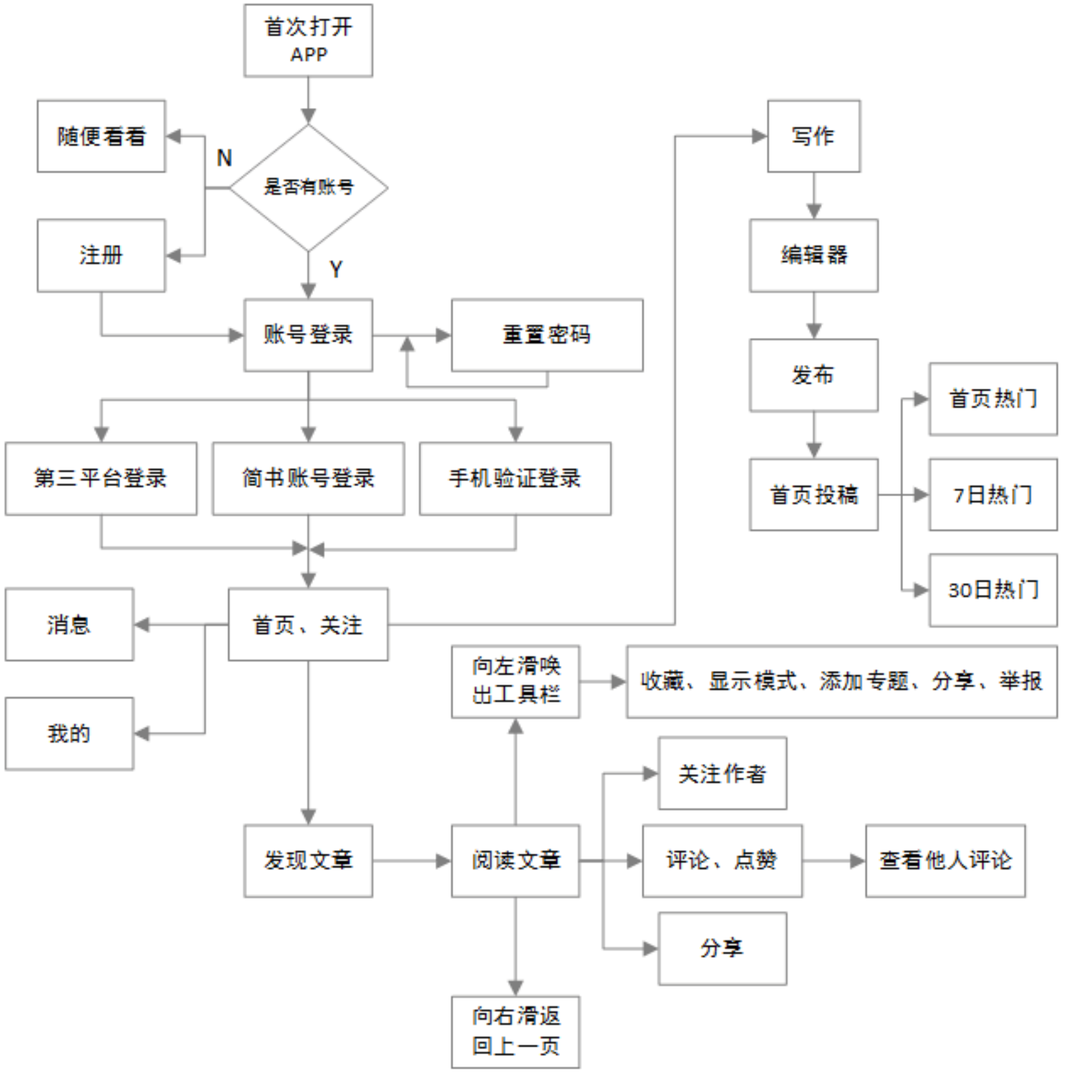
第二部分：程序设计的工程角度

- 实际问题经常是这样的：

请为P大学开发一套“学生信息管理系统”
请为A公司开发一个网站
有个idea想做个手机App
…

- 如何建模成计算问题?


简书APP
发现
Banner
专题
搜索
热门文章
首页投稿
关注
简友圈
关注列表
添加关注
搜索关注
关注设置
消息
评论
简信
投稿请求
喜欢
关注
打赏
其他
我的
个人资料
公开文章
私密文章
我的收藏
我的专题
我的文集
我的钱包
其他设置


首次打开APP
是否有账号
N
Y
随便看看
注册
账号登录
重置密码
第三平台登录
简书账号登录
手机验证登录
首页、关注
消息
我的
写作
编辑器
发布
首页投稿
首页热门
7日热门
30日热门
向左滑唤出工具栏
收藏、显示模式、添加专题、分享、举报
发现文章
阅读文章
关注作者
评论、点赞
查看他人评论
分享
向右滑返回上一页


NetInSight
NetInSight
有你我眼里的互联网，更有产品人探索的表达
写了10374字 获得了71个喜欢
编辑个人资料
输入按钮
动态
文章
更多
文章 (10)
最新文章
NetInSight 10-21 16:57
一张图，可以否定了人人都是产品经理
209次阅 • 0评论 • 7喜欢
NetInSight 10-20 23:13
为何产品思考，我爱用脑图表达
75次阅 • 0评论 • 0喜欢
编辑个人资料
136****3714
未绑定
未绑定
未绑定
豆 未绑定
G+ 未绑定
社交帐号是否显示在主页
重置密码
点击进入重置密码页
重置密码
为了你的帐户安全,请选择下一种方式进行身份验证
使用当前密码验证
使用手机号验证
136****3714
使用邮箱验证
pan****@163.com
点击进入手机验证页
并立即发送验证码
重置密码
请输入验证码
重发 47
请输入新密码
请再输入一次新密码
完成
需完成验证码、
密码、确认密码

需求经常会改变

- 场景：
  - P大信息系统：一开始没说需要做国际生
  - 后来有国际留学生：
    - 外文支持
    - 留学生特别的管理方法（例如学制，学生角色等）
- 程序需要设计成能应对经常发生的需求改变

需求经常会重复

- 场景：
  - 同一个团队又给T大写了学生信息管理系统，如何让修改尽量少?
  - 团队最近又接手了为A公司做员工信息管理系统
  - 经常需要使用的功能常被写成类库来发表
- 程序功能模块重叠需要复用，这种复用不是复制-粘贴，需要支持一定的灵活性

复制-粘贴实现复用有什么问题?

团队协作性与可维护性

- 场景：
  - 维护：给P大写完了系统，出bug需要可以容易维护
  - 协同开发：为追求效率，一个项目经常需要很多人同时进行开发
- 需要将整个任务切分成相对独立的小模块

程序设计的范式

- 面向对象：为了有效进行复杂需求建模，并且可复用、可灵活修改
- 还有其他编程范式，各自有不同的诞生背景/侧重
  - 面向过程编程
  - 函数式编程
- 另外，范式还可以体现在编程框架下，不拘泥于语言 （MapReduce，Qt等）

程序设计范式之间的关系

- 范式是一种“管理学”，不同范式从解决问题的“能力”上没有“本质区别”
  - 不论什么编程语言，能力都是图灵机
  - 任何X语言写的程序的功能也可以用Y语言实现
  - 然而实现的难度可以有很大差别：汇编 vs Python脚本
- 应根据解决问题的需要，灵活选择设计范式（语言/工具等）

第二部分大纲：面向对象编程（C++）

- C++的面向对象语言特性
  - 类/对象
  - 继承/多态
  - 模板类/模板方法，STL等
- 设计模式选讲
  - 案例 + 经典设计思路

部分内容基于 Design Patterns: Elements of Reusable Object-Oriented Software; Head First Design Patterns

课程计划

- 每周 周三，单周 周五上课
  - 单周：算法专题（主要讲“现代”算法）
  - 双周：经典算法选讲和面向对象编程（最后几次课可能是客座讲座）
- 现代算法：介绍性为主，主要理解算法过程、会实现，以及直觉上的设计原理
- 经典算法与面向对象的内容与普通班基本保持一致

课业负担

预计~15次，取决于进度

- 每1 - 2次课后布置编程小作业/实验报告，每个作业计1的权重
- 面向对象会有一个大作业（与普通班类似），计5的权重
- 没有期中考试

会有一大部分题目与普通班同步（只做这些可优秀）
但想考90+需要做出与现代算法有关的内容

- 期末上机考试：不专门考察面向对象，主要考察算法（经典和现代）
- 分数：平时作业50%，期末考试50%

可能最后根据情况进行调整，但是不会让你的分数更低

- 每个作业给至少3周时间；所有作业按照权重按比例分配50%的分数

选课指导

- 内容上基本涵盖普通班，但可能较为简略，重点放在“现代”算法的介绍上
- 实验班没有期中考试，对于C++面向对象也不在考试中考察
- 基础要求上较普通班高，但多数内容并不需要额外修课程就可以学
- 竞赛生/有算法基础学起来容易一些；无基础的同学可以试听1、2节课感受一下

复习知识点

- 什么叫做算法的近似比？针对最小化和最大化问题都是如何定义的?
- 如何线性时间求平面点集直径的2-近似?