

数据结构与算法 (Python)

课程引言

谢正茂 webg@PKU-Mail

计算机学院数据所

February 24, 2025

目录

- 课程定位
- 编程语言和教材的选择
- 考核方式
- 结语

计算机科学的领域

3 计算机科学的领域

3.1 理论计算机科学

3.1.1 数据结构和算法

3.1.2 计算理论

3.1.3 信息论与编码理论

3.1.4 编程语言和编译器

3.1.5 形式化方法

3.2 计算机系统

3.2.1 计算机体系统结构与计算机工程

3.2.2 操作系统

3.2.3 并发、并行与分布式系统

3.2.4 计算机网络

3.2.5 计算机安全和密码学

3.2.6 数据库

3.3 计算机应用技术

3.3.1 计算机图形学

3.3.2 科学计算

3.3.3 多媒体技术

3.3.4 人工智能

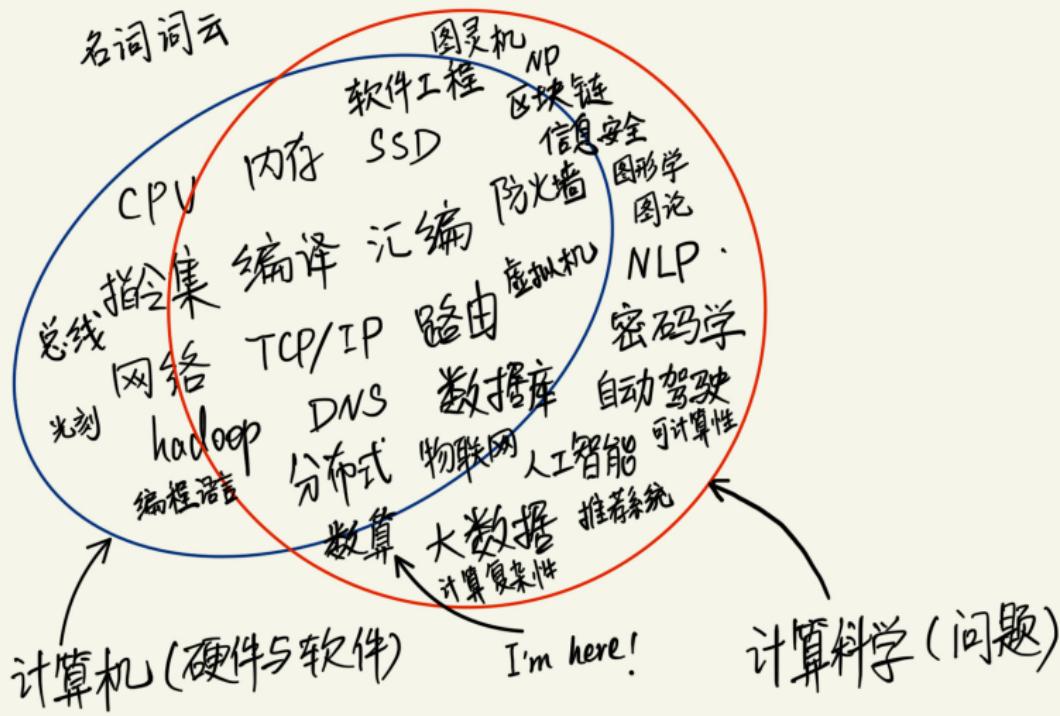
3.4 软件工程

- 计算机科学并不只是关于计算机，就像天文学并不只是关于望远镜一样。
- 设计、部署计算机和计算机系统通常被认为是非计算机科学学科的领域。
- 计算机科学被认为比其它科学学科与数学的联系更加密切，一些观察者说计算就是一门数学科学。



WIKIPEDIA

课程定位



- 这门课讲些什么，讲到什么程度？
- 目标：希望同学们学完之后有哪些收获
 - 写程序处理分析数据的能力，让“计算机”成为强大助力。
 - 进一步学习人工智能/机器学习，很好的起点。
- 计算机的影响和使用已经深入到了各行各业。
 - 计算机七十年来的发展与近十年的爆发
 - 信息革命：数字化、大数据时代、摩尔定律
 - 中国在这场革命中的地位与机遇

课程主要内容

● 课程基本框架

- Python 入门
 - 算法基本概念与复杂度
 - 线性表
 - 递归与动态规划
 - KMP 算法
 - 排序与查找
 - 树及算法
 - 图及算法
-
- 比《计算概论》讲的窄、但更加深入算法。
 - 按照教材的脉络，但引入了大量新的内容



“算法”课程从限选变成必修

- 需求激增，选课人数翻番！
- 近年来发生的大数据、互联网+，给这个领域带来了大量的需求和工作岗位。
- 大学之后找工作/创业都是不错的选项。
 - 除研发类的少数岗位以外，“外院”的同学都能胜任。
 - 算法类、产品，“外院”同学还有某些优势。
- 继续做研究：数据驱动的研究。
 - 在大数据的时代背景下，不仅是“理工科”，“文科”也需要用数据说话。
 - 最难想象的是，在“考古领域”大数据都能带来重大的发现
 - 写程序获取、处理、分析数据成为“文理工”共同的基础技能。
- 计算机专业考研的话，请出门右转选“C”班。



目录

- 课程定位
- 编程语言和教材的选择
- 考核方式
- 结语

- 程序是计算机命令的序列。
- 程序设计语言是对数据结构和算法的表达。
- PASCAL 语言 ==> C/C++, JAVA, Python
- 高级语言 -> 汇编语言 -> 机器语言
- 高级语言里面也分谁更“高级”，高级是抽象的层次。
 - “高级”与否无关好坏，根据具体的工作进行选择。
 - 每种语言都有自己流行的领域。在一个领域越流行，开源的资源越多。
 - 抽象的层次高，屏蔽了底层的细节，开发起来省力；但有些功能实现不了。
 - 如果计算机专业的话，最好对 C/C++ 有所了解。
 - 如果从事硬件开发的化，需要了解汇编。

程序的开发效率与运行效率

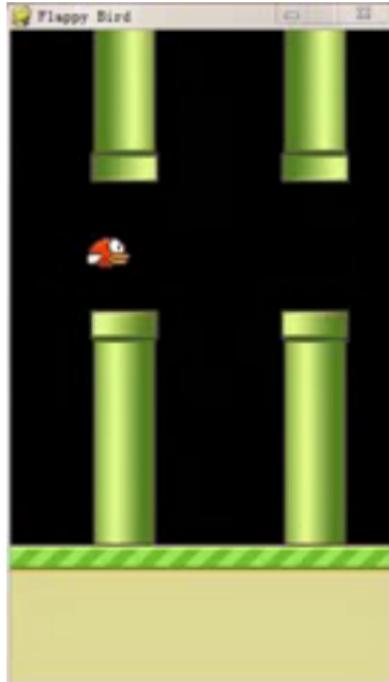
- 开发效率: 's/typo/type/g'
- 运行效率: curl 1998 至今, 70 亿安装量 ←
- Python vs C/C++ vs Assembly ←
- 绝大多数情况开发效率优先, 除非你的程序
 - 很少改动
 - 大量运行
- 运行效率够用就行。不适用于本课程!!!

为什么选 Python?

- 代码短小精悍，干净整洁
 - 没有变量声明，不需要花括号 begin/end，也没有分号，比 java 短 80%，比 C 短 98%
- 解释执行，上手就玩，编程小白福音
 - 不用焚香沐浴安装 GB 级别的开发环境 compile/build，可以随问秒答，边玩边改
- “包装内附带电池”
 - 自带大量运行库，网络、数据库、图形图像、GUI、压缩加密一应俱全，几行代码建网站
- 功能无比强大，开发左右逢源，最酷的网络应用都是用它
 - Google/Youtube/Instagram/豆瓣……，NASA 也用它哦
- 搞大数据和 AI 的人们也爱它
 - 有各种面向大数据处理的数据模型、数值分析、机器学习、空间分析等 Python 工具随时恭候

Python 坐稳人工智能时代的头牌语言

- 机器学习“全家桶”scikit-learn
- Google 开源的 AI 系统 Tensorflow
- Python 可以调用 C++ 的代码
- 160 行 Python 代码可以让 AI 从游戏视频中学习玩 Flappybird

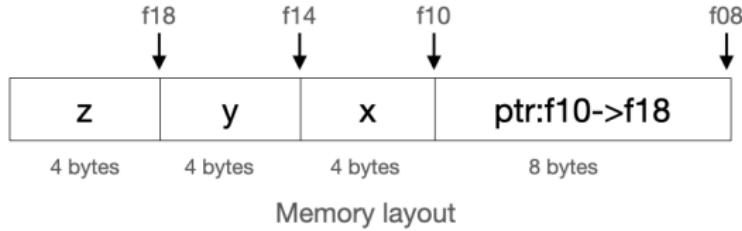


C 与 Python 的“非全面”比较：关于指针

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int z, y, x;
6     int *ptr = &x;
7     printf("ptr is %p\n", ptr);
8     printf("ptr's address is %p\n", &ptr);
9     printf("sizeof(x) is %lu\n", sizeof(x));
10    printf("sizeof(ptr) is %lu\n", sizeof(ptr));
11    *ptr++ = 5;
12    *ptr++ = 6;
13    *ptr = 7;
14    printf("ptr is %p\n", ptr);
15    printf("ptr's address is %p\n", &ptr);
16    printf("x = %d\n", x);
17    printf("y = %d\n", y);
18    printf("z = %d\n", z);
19    return 0;
20 }
```

```
(base) rmbp13:code xiezhengmao$ ./pointer
ptr is 0x7ffecbc8f10
ptr's address is 0x7ffecbc8f08
sizeof(x) is 4
sizeof(ptr) is 8
ptr is 0x7ffecbc8f18
ptr's address is 0x7ffecbc8f08
x = 5
y = 6
z = 7
```

- 11-13，绕过变量直接访问内存
- 在地址空间中可以随意移动
- 把（虚拟）内存暴露给程序员，因而支持一些强大的功能。
- 同时也是坑人王！包揽 95% 的程序 bug
- 地址越界、野指针、程序后门
- 强制人像机器一样考虑问题



一个常见的 bug

```
diff --git a/main.c b/main.c
index b6742a9..c5ae788 100644
--- a/main.c
+++ b/main.c
@@ -4,7 +4,7 @@ const char* foo = "David";

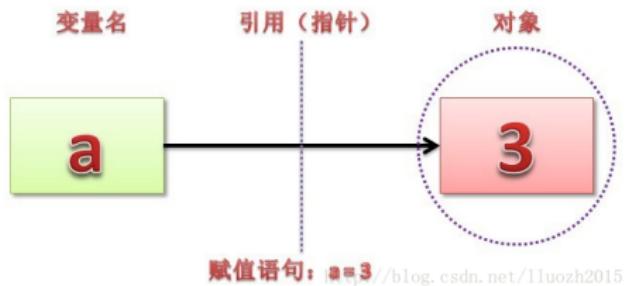
int main()
{
-    char* name = (char*) malloc (strlen(foo));
+    char* name = (char*) malloc (strlen(foo) + 1);
    strcpy(name, foo);
    return 0;
}
```

- `strcpy()` functions copy the string src to dst (including the terminating '\0' character.)
- '\0' 写在了没有分配的空间上
- 地址越界 bug: 不一定马上出错/在不同的平台上表现不一样
- 最难找的 bug: 潜伏期长, bug 和程序 crash 在不同地方
- 专门为他开发 debug 工具: <https://dmalloc.com/>
- 用起“指针”来如履薄冰

C 与 Python 的“非全面”比较：指针的使用

- 争论：Python 中有没有指针？
- 感觉不到指针 vs. 所有可赋值的东西都是指针
- 变量只是指针/引用/标签：identity of object
- 封装：把对指针的直接操作藏起来（傻瓜化）
- `id()` 实际上就是内存地址，但语言极力避免用户直接操作地址。

```
>>> l=[1, 2, 3]
>>> ll=l
>>> ll[1]='David'
>>> l
[1, 'David', 3]
>>> id(l), id(ll)
(140381553307392, 140381553307392)
>>> import _ctypes
>>> _ctypes.PyObj_FromPtr(140381553307392)
[1, 'David', 3]
>>> hex(id(l))
'0x7fad209e2f00'
```



Python 哲学

- rule 3: Simple is better than complex.
- rule 7: Readability counts.
- 把所有东西都拿出来 vs. 尽可能的“藏”起来
- 专注于问题本身，而少管与“计算机”相关的东西。
- 与之相反，“体系结构”专业专门研究与“计算机”相关的东西。
 - 从逻辑门开始，研究寄存器、加法器、乘法器、指令集、CPU、内存、南桥北桥、硬盘，.....
 - 体系机构教研室，开发了国产处理器：北大众志；军用保密芯片。



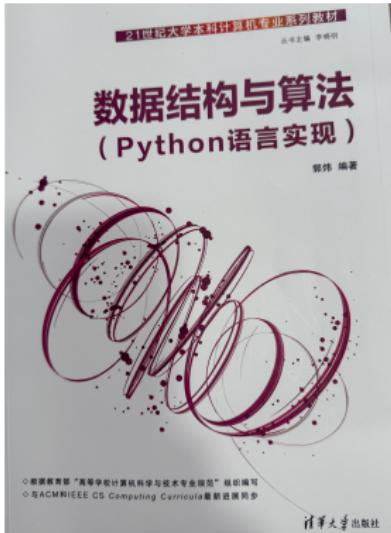
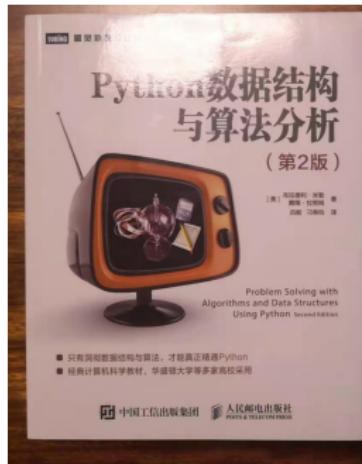
为什么 C 会那样“不友好”?

- 与底层硬件 (CPU) 靠的最近：C 代码和汇编指令有简单的对应关系，可以手工翻译。
比如说：指针操作、程序跳转 Goto 语句
- 新的硬件出来，最先支持的高级语言肯定是 C
 - 硬件：奔腾、AMD、高通枭龙、华为麒麟、苹果 M1、北大众志、龙芯
 - 指令集：x86、MIPS、Sparc、Alpha、ARM
- Python 语言也是由 C 实现的¹。
- Windows、MacOS、Linux 的内核都主要用 C 实现的
- 隔壁班同学提问：既然 C++ 包括了 C，我们为什么不直接讲 C++?
:) 其实它们的位置完全不一样。而 Python 和 C++ 的位置更近。
今年来 C++ 快速更新，C++11/14/20 大量借鉴了 Python 的一些特性，而 C 保持相对稳定。

地址	机器指令	汇编指令
0x0804857a	894442404	mov dword [esp + 0x4], eax
0x0804857e	c70424b28604.	mov dword [esp], 0x80486b2
0x08048585	e8eafdffff	call sym.imp.scant

¹<https://www.python.org/downloads/source/>

主要教材与参考书



- 教材 pythonds²，线上版本比印刷版本内容更加新。
- 源代码：<https://github.com/RunestoneInteractive/pythonds>
- 新书：知识覆盖面更广，尤其是算法部分；高标准的实践性
 - 天猫：<https://detail.tmall.com/item.htm?id=891563544947>

²<https://runestone.academy/runestone/books/published/pythonds/index.html>

如何学习一门程序设计语言?

● 教程 vs. 手册

- 手册包含了语言完整的特性，面面俱到，并详细阐述技术细节；一般都是大部头。
- 教程教人快速上手，丰俭由人，有所偏重；一般都是中短篇幅。
- 学会之后需要有机会经常练习，否则忘的很快。
- 看书之外，使用开发工具自带的帮助，随用随查，更方便/更常用。
- 初学的时候，一两页的“Cheat sheet”也许能够解决你大部分的需要。

- 天网搜索引擎

- 北大天网由北京大学网络实验室研究开发，是国家重点科技攻关项目“中文编码和分布式中英文信息发现”的研究成果。北大天网于 1997 年 10 月 29 日正式在 CERNET 上向广大互联网用户提供 Web 信息搜索及导航服务，是国内第一个基于网页索引搜索的搜索引擎。

- Web Infomall

- 中国互联网页信息博物馆，从 2002 年开始对中国的互联网网页进行增量搜集、存储、展示。网络爬虫每天的数据搜集能力达到了三千五百万网页。

- 区块链（联盟链）的应用场景设计

目录

- 课程定位
- 编程语言和教材的选择
- 考核方式
- 结语

- 满分 100 分
 - 平时作业: 20%
 - 平时小测: 10%
 - 期末机考: 30%
 - 期末笔试: 40%
 - 以上各项教师保留 5 分的调整权利
- 刷题、机考网站: <http://xzmdsa.openjudge.cn/>
- 数算这门课考核分成两个部分：理论知识和实际编码能力。实际编码能力需要大量刷题，提高不是一朝一夕之功；理论相对来说更容易，认真上课、复习都能得到不错的分数。
- 附加分 10 分
 - 课堂上能够指出老师的问题，每次有不超过 0.5point 额外加分
 - github 在线协作加分（参与也能获得锻炼提高）
 - github 地址 <https://github.com/Patrickxzm/dsa2020>
 - 镜像地址 <https://gitee.com/patrickxzm/dsa2020>
- 最重要的期末笔试，所有班统一命题，本班课上讲的不一定考

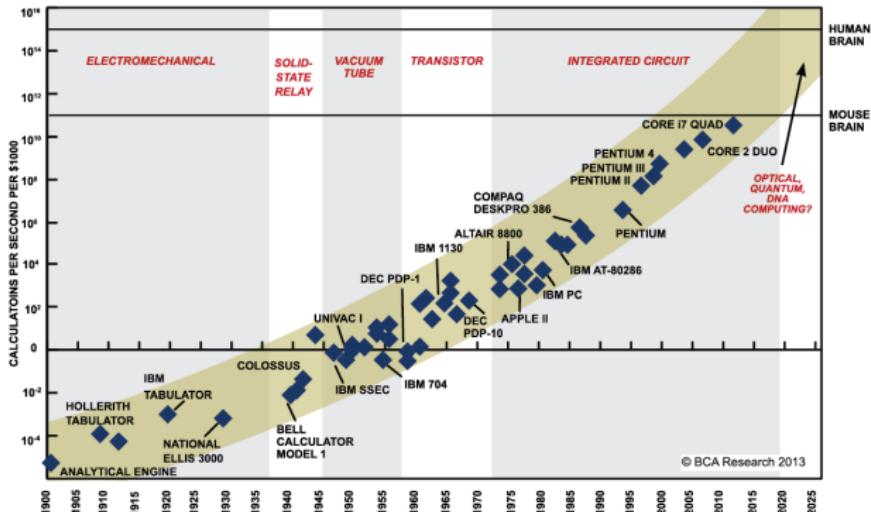
- 如果上机课与我的其他课程冲突了怎么办?
 - 期末机考安排在最一次上机课时间（6月5日，16周）
 - 需要确保能够参加，没有重考补考的可能
- 上机课助教会来机房，提供线下的答疑时间；更多的讨论和答疑在github 上进行。
- 机考不能携带自己的电脑，答题在机房的电脑上进行，因此需要对它的编程环境足够熟悉。

目录

- 课程定位
- 编程语言和教材的选择
- 考核方式
- 结语

摩尔定律 (Moore's law)

- 集成电路上可容纳的晶体管数目，约每隔 18 个月便增加一倍。
- 微处理器的性能每隔 18 个月提高一倍，或价格下降一半。
- 相同价格所买的电脑，性能每隔 18 个月增加一倍。
- 1996 年以来的摩尔定律由英特尔 (Intel) 创始人之一戈登·摩尔提出 (两年一倍)；英特尔首席执行官大卫·豪斯 (David House) 改为“18 个月”。



- 数字化：数据的产生
 - 趣闻：马斯克说，美国的纸质退休档案在废弃矿井中保存
 - 最新提法：产业数字化，数字产业化
- 大数据：数据的存储、流动、汇集与应用
 - 网格，云：数据、算力、服务的随处可达
- AI：数据应用下结的，被大数据催熟的大瓜
- 推波助澜，后面都有摩尔定律的影子
- 中国刚好赶上了最近三十年的这一波浪潮，有了猛烈的发展
 - 整个民族要增强信心，继续发展
 - 精英们要开阔眼界，看到差距，不可盲目自大
 - 虽然西大最近很拉胯，但人家家底厚！