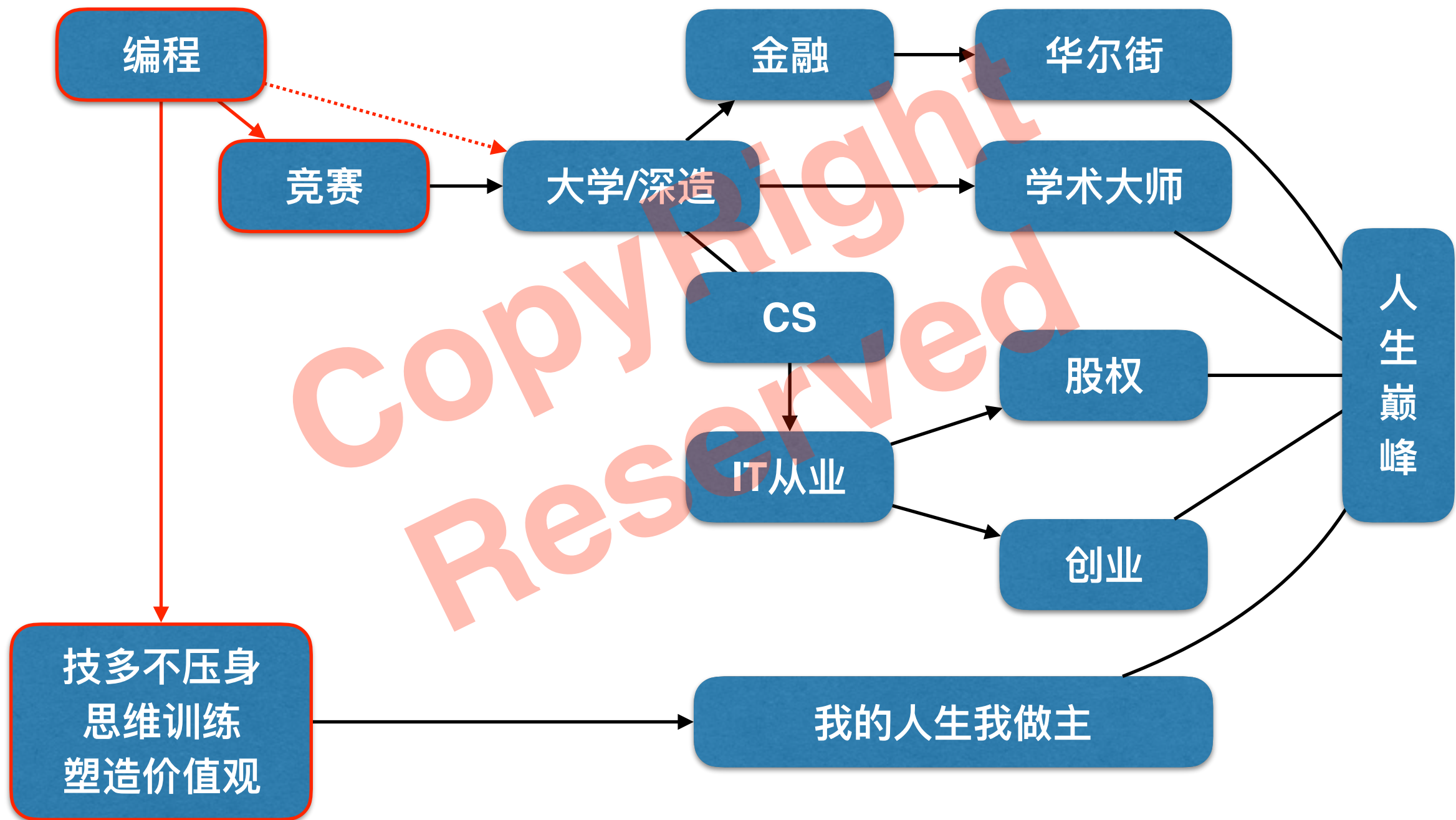


# CS100 算法入门

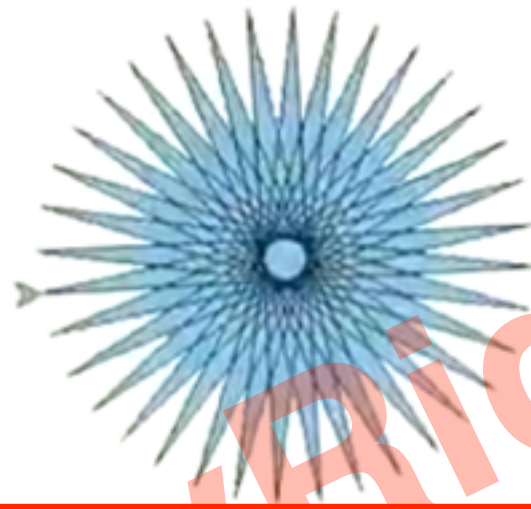
阳春白雪

# 人生规划：为什么要学编程/学算法

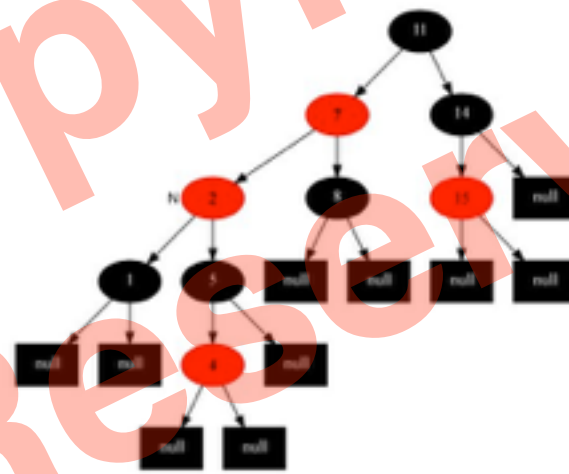


# IT研发的层次与能力结构

展现层(前端/UI)



逻辑层  
(算法/工程/架构)



存储层



# 怎样学好算法



## • 黄金法则

- 建模(灵魂) > 算法(骨骼) > 语言(细胞)
- 兴趣是最好的老师
- 练习自学/自己排查问题, 你会受益终生
- 利用网络, 不要闷头空想

# 语言的选择

---

## 江湖上流传的梗：

程序员：美女，能请你看电影吗

女神：如果你能让这个论坛的人都吵起来，我就跟你去

程序员：（发帖）PHP是全世界最好的语言

某论坛真的就炸锅了，各种吵架.....

女神：我服了，我们去看电影吧

程序员：不行，我一定要说服他们PHP是全世界最好的语言

## 本课程对语言的定位：

- 1.是工具不是目的
- 2.原则上算法用任何语言都能实现
- 3.特定场景下，某些语言会有一定优势
- 4.没有万能的(高级)语言

容易上手    python

功能强大    C++

性能好    C++

生态好    Java

大数据挖掘    R

.....    .....



# 为什么选择C++

竞赛规定语言之一

经典：历史悠久，功能强大  
(前身C语言堪称高级语言鼻祖)

性能好

有强大的控制力(上乘武功)

大量使用者，生态圈完善

看得懂的都是老司机



# 课程内容（宏观）

---

1-1	经验分享，宏观概况
1-2	C++语言入门
2	排序
3	分治法
4	树
5	递推+回溯法
6	堆+位运算
7	散列表
8-1	高级话题
8-2	总结

- **与竞赛的关系（你们最关心的）**

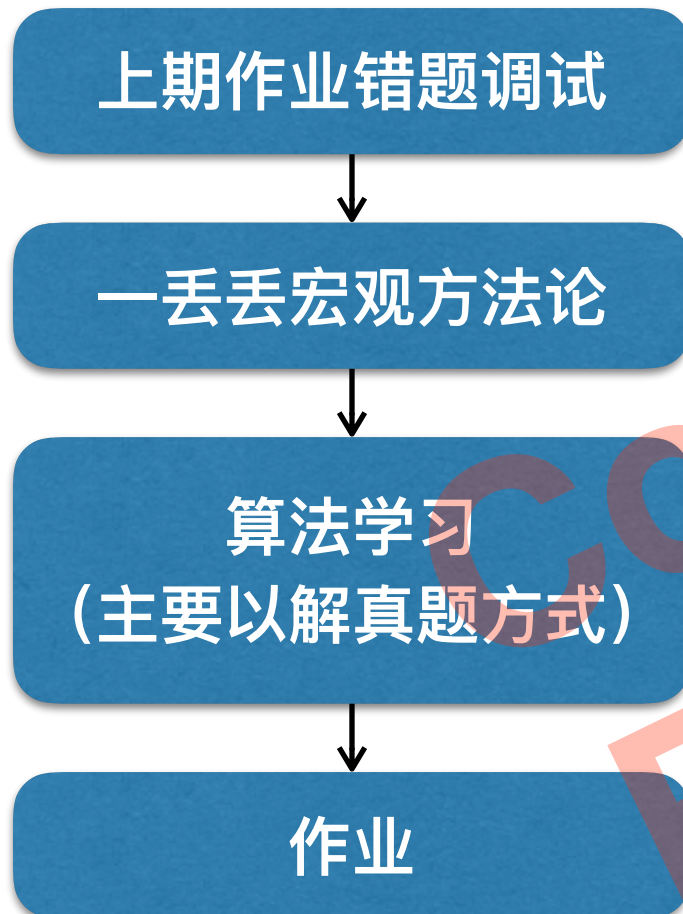
- 所学算法都是竞赛有用的
- 非应试，不包含应试细节/技巧（你们可以找陈老师开个专题讲座🤪）
- 不区分普及/提高组，以算法递进为纲

- **其他内容**

- 穿插少量IT工程/互联网相关内容（与竞赛无关）
- 不涉及纯学术探讨
- 语法只讲一节课（语言只是工具），遇到语法问题/编译错误尽量自己解决

# 课程内容（微观）

---



- 算法属于逻辑层，没有CS001的观赏性和EE100的趣味性（这可是你们自己选的哦，你们很勇敢😘）
- 建模是算法的灵魂。所以你不要只关注算法本身，更重要的是体会怎么为具体问题选择合适的算法
- 作业有很多开放性问题，不要求全部做。做得好会有红包（每周都会有^^）



# 学习模式



金字塔式  
(夯实基础, 循序渐进)



冰川式  
(松散堆积, 渐压成型)

→ 符合大脑的学习模式

→ 有一些挑战性, 但也能激发潜力

**NOTE: 不是无限堆积!**

# 你该怎么学（重要！）

---

- 体会建模的思路（灵魂，只可意会，所以我要讲真题）
- 听不懂怎么办？先撑一下，课下再回味
- 练习，练习，练习（但不要熬夜）
- 刷题库？可以，但不要以AC为目的，要理解解法
- 有问题怎么办？先尝试自己解决（查书，问度娘，调试。这三项都是你以后必备的技能）。解决问题的能力比答案本身更重要。至少试三次，不行再来问我
- 作业做不完怎么办？本来就不期望都做完，量力而行

压力山大？要超过别人总是要跳一跳的。你可以的



# 其他纪律

- 怎么提问

不要这么问	应该怎么问
老师，这题怎么做？	这题我的思路是...有什么问题
老师，我这代码哪里不对？	你贴给我的代码必须加详细注释，说明每一步是做什么
老师。报错了	根据错误提示自己Baidu



给你一个眼神  
自己去体会

- 课堂纪律

- 我讲为主，听讲的时候电脑不要开
- 如果你觉得我讲的太慢，你已经理解了，举手说。可以让你讲一段（鼓励。我这不是挑衅啊）
- 中途实在不想学，可以退出（陈老师不要抽我(\*'д` )）。记住**不会算法生活一样可以很精彩**（我是业界良心吧）

# 作业

---

## 1.想一想，你为什么来学编程/学算法

(提示：如果不知道，问一问你们的家长😏)



## 2.去查一查，你能找到多少种现存的编程语言？

(找到最多的3位有红包^^)

### • 红包怎么发？

- 作业截止每**周三22:00**，晚交不参与红包
- 提交到**stem111@qq.com**，标题格式：**cs100第X周作业-姓名**。非编程题统一写成一个docx附件。编程题提交cpp文件附件，文件名会在题目里注明。有任何问题/吐槽可以写在正文里😏
- 每个红包符号10块，所有获得资格的一起分（人人有）