

CS专业打怪升级指南 —— 大学经验分享 演讲报告

演讲者: 李和平 (计算机科学与技术专业 大四)

对象: 计2501-3班全体同学

时间: 2025年12月

场合: 晚自习学习经验分享会

📌 前言：为什么需要这份“攻略”？

各位学弟学妹，晚上好！

首先，非常感谢王兴亚老师提供这次交流的机会，也感谢大家抽出宝贵的晚自习时间。

我知道，现在的你们可能正在经历以下“新手村困境”：

- 🧑 刚考完四六级，还没来得及喘息，期末考试已经迫在眉睫
- 📚 高等数学的极限、微积分让人头秃
- 💻 C++的指针像一团乱麻，满屏红色Error不知所措
- 🤔 大学和高中到底有什么不同？为什么感觉“努力”了却没有回报？

迷茫是常态。我也曾是那个对着屏幕发呆的大一新生。

今天，我不想讲空洞的大道理，只想和大家聊聊：**如果能“读档重来”，我会怎么玩这场大学游戏。**

第一章：🎮 大学 vs 高中 —— 游戏规则彻底变了

1.1 最大的不同：从“被动刷题”到“主动探索”

维度	高中模式	大学模式
目标	唯一目标：高考分数	多元目标：GPA、技能、项目、人脉、视野

维度	高中模式	大学模式
学习方式	老师划重点，题海战术	自学为主，老师是引路人，不是保姆
时间管理	学校安排到分钟	大量自由时间， 自律者自由，放纵者迷失
评价标准	分数排名	综合能力（代码、项目、竞赛、实习、表达）
犯错成本	一次考试失误可能影响人生	试错成本低，但要学会从错误中成长
信息获取	老师告诉你一切	信息差是最大的敌人 ，需要主动搜索

1.2 关键认知转变

高中是"标准答案"的游戏，大学是"开放世界"的探索。

1. **没人逼你了** —— 上课可以玩手机，作业可以抄，考试可以水过。但四年后，市场会给你一个"残酷的分数"。
2. **努力的方向比努力本身更重要** —— "伪勤奋"是大学最大的陷阱。坐在图书馆玩手机 ≠ 学习。
3. **成绩只是一部分** —— GPA很重要（保研/出国/求职的门槛），但不是全部。项目经验、实习经历、技术深度同样关键。
4. **社交不是浪费时间** —— 你的同学，可能是你未来的合伙人、同事、或贵人。学会合作，学会表达。

第二章：💡 技能树（一）—— 内功心法：数学基础

2.1 为什么要学高数/线代/离散/概率？

很多同学问："高数买菜又用不到，学它干嘛？"

真相：这些课程是你未来解锁高级技能的前置条件（Pre-requisites）。

基础课程	未来解锁的高级技能	重要性
微积分	机器学习、深度学习、AI算法中的梯度下降、反向传播	★★★★★
线性代数	计算机图形学、游戏开发、推荐算法、图像处理	★★★★★
离散数学	数据结构、算法设计、密码学、逻辑思维	★★★★★
概率论与统计	数据科学、大数据分析、A/B测试、贝叶斯推理	★★★★★

2.2 数学学习的正确姿势

✗ 误区：

- "上课听不懂就算了，考前突击"
- "数学太抽象，背公式就行"
- "以后又不做科研，学这些没用"

✓ 正解：

- **即使听不懂，也要硬着头皮刷题**，保持"题感"。很多概念是"做着做着突然懂了"。
- **出来混迟早要还**：大一欠下的数学债，大三学机器学习、计算机视觉时都要加倍偿还。
- **理解 > 背诵**：不要死记公式，理解推导过程。考试时能现场推导的人，不会被"变形题"难倒。

2.3 期末复习策略（高数篇）

针对刚考完四六级、还有不到两周期末的你：

1. **课后习题 > 课外难题**：老师出题很大概率参考课后习题或往年真题。
2. **背熟基本积分公式**： $\int x^n dx$ 、 $\int e^x dx$ 、 $\int \frac{1}{x} dx$ 等必须秒出。
3. **真题至上**：找近3年的期末真题，反复刷，搞懂每一道错题。
4. **时间分配**：高数计算量大，建议每天至少2-3小时集中复习。

第三章： ✨ 技能树（二）—— 招式修炼：编程语言

3.1 编程学习的核心法则

"Talk is cheap, show me the code."

—— Linus Torvalds (Linux之父)

翻译过来就是：少说废话，代码说话。

3.2 编程学习的误区与正解

✗ 误区：

- 背代码、抄代码、只看书不动手
- 背代码、抄代码、只看书不动手
- 遇到 Bug 就百度复制粘贴，不求甚解

- 觉得"反正有AI了，不用自己写代码"

✓ 正解：

- **关掉课本，打开IDE**：VS Code、CLion都是你的练武场。
- **红色的 Error 是你最好的老师**：每一个报错都是一次学习机会。学会读错误信息！
- **Debug能力 > 写代码能力**：会写代码的人很多，能高效Debug的人才是高手。
- **搜索技巧**：多用 Google / Stack Overflow / AI工具，少用百度（技术内容质量差异巨大）。

3.3 C++指针——新手最大的坑

示例：空指针解引用（Segmentation Fault）

```
// ✗ 错误代码
#include <iostream>
using namespace std;

int main() {
    int* ptr = nullptr;
    *ptr = 2077; // 致命错误：向空指针写入数据
    cout << "Val: " << *ptr;
    return 0;
}
// 运行结果：Segmentation fault (core dumped)
```

```
// ✓ 正确代码
#include <iostream>
using namespace std;

int main() {
    int data = 2077;
    int* ptr = &data; // 修复：让指针指向合法内存
    cout << "Val: " << *ptr;
    return 0;
}
```

指针学习要点：

1. 指针就是"地址"，理解内存模型是关键
2. 使用指针前，必须确保它指向合法内存
3. 动态内存分配（new / delete）要配对使用，否则内存泄漏

3.4 期末复习策略 (C++篇)

1. **死磕重点**: 指针 & 内存管理、结构体、文件操作、基本算法（排序、查找）
2. **老师画的重点 = 考点**: 不要自己发挥，老师说什么考就复习什么
3. **上机练习**: 编程是“手艺活”，光看不练假把式。把往年编程题都敲一遍
4. **理解错误信息**: 知道 Segmentation fault 、 未定义标识符 等常见错误的原因

第四章： 装备库 —— 工欲善其事，必先利其器

4.1 VS Code (现代IDE)

为什么要用 VS Code?

- 轻量、启动快
- 插件生态极其丰富
- 跨平台 (Windows/Mac/Linux)
- 免费开源

必装插件：

- C/C++ : 代码补全、调试支持
- Prettier : 代码格式化
- Error Lens : 错误信息直接显示在代码行内

 警告：

- 拒绝只用记事本写代码
- 尽早抛弃古老的 VC++ 6.0 (20年前的工具了！)

4.2 AI辅助工具 (Copilot / DeepSeek / ChatGPT)

正确定位：AI是你的24小时私教，不是替你写作业的枪手。

Prompt类型	示例	效果
 坏Prompt	"帮我写个冒泡排序"	你失去了思考能力，下次还是不会

Prompt类型	示例	效果
好Prompt	"像给5岁孩子讲故事一样解释这段指针代码"	你理解了底层逻辑
好Prompt	"这段代码为什么报错? 请解释原因并给出修复思路"	你学会了Debug

黄金法则：

- 先自己思考，卡住30分钟以上再求助AI
- 让AI解释原理，而不是直接给答案
- 验证AI的输出，AI也会犯错

4.3 GitHub（程序员的军火库）

为什么一定要用 GitHub？

1. **代码托管**：你的代码有个安全的"云备份"
2. **版本管理**：告别 `final_doc.doc`、`final_doc(1).doc`、`final_doc_真的最后版.doc` 的噩梦
3. **开源宝藏**：全球最大的开源社区，探索无数优质项目
4. **求职名片**：你的 GitHub Profile 就是你的技术简历
5. **团队协作**：解锁"多人联机模式"，Git 是团队协作的基石

大一就该做的事：

- 注册 GitHub 账号
- 学会基本命令：`git add`、`git commit`、`git push`、`git pull`
- 把自己的课程作业、小项目上传（形成积累）

第五章：🔥 当前BOSS战 —— 期末生存指南

5.1 当前状态评估

- **刚考完四六级**：精神上可能有些疲惫，但别松懈
- **距离期末**：< 2周
- **目标**：存活并高分（不挂科是底线，高绩点是选择权）

5.2 复习优先级矩阵 (ROI思维)

核心思想：不要平均用力，死磕高回报内容。

内容	投入产出比	优先级
指针 & 内存管理	★★★★★	● 最高
微积分计算	★★★★★	● 高
程序设计基础	★★★★★	● 高
极限	★★★	● 中
常见语法细节	★★	● 低

解读：

- 指针是C++的灵魂，也是考试的重灾区，必须拿下
- 微积分计算题分值高，公式要熟练
- 纯理论定义的死记硬背，投入产出比最低

5.3 时间管理（倒推法）

步骤：

1. 列出所有考试科目和时间
2. 倒推每天需要复习的内容
3. 精确到小时，做成“甘特图”
4. 每天晚上检查完成情况，动态调整

示例（假设还有14天）：

Day	上午	下午	晚上
Day 1	高数：极限复习	C++：指针专题	刷高数真题
Day 2	高数：微分	C++：内存管理	刷C++真题
...

5.4 高效复习的秘诀

1. 拒绝自我感动

- "坐在图书馆玩手机8小时" ≠ 学习
- 效率 > 时长
- 专注2小时 > 摸鱼8小时

2. 刷真题，刷真题，刷真题！

- 近3年真题是最好的复习资料
- 搞懂每一道错题，整理错题本
- 出题老师的套路是有迹可循的

3. 睡眠 > 熬夜

- 熬夜复习的效率极低，第二天还会犯困
- 保证每天6-7小时睡眠
- 考前一天绝对不要通宵

4. 心态建设

- 适度紧张是正常的，别焦虑到影响发挥
- 不挂科是底线，高绩点是选择权
- 一次考试不能定义你的人生

第六章： 主线任务 —— 毕业硬指标

6.1 英语四六级 (CET-4/6)

为什么重要？

- 很多大厂招聘要求"六级425+"甚至"六级480+"
- 保研/出国对英语成绩有硬性要求
- 英语是程序员的必备技能（文档、开源社区、面试）

策略：

- 大一/大二记忆力最好，一鼓作气拿下
- 不要拖到大四！
- 四级过了立刻冲六级，趁热打铁

6.2 PU学分（第二课堂）

毕业门槛，拒绝临毕业才疯狂补分。

类别	要求	建议
总实践学时	150分	平时多参加活动，轻松达标
校园文化活动	60分	讲座、社团活动等
思政社科	20分	观看相关课程/讲座
公益服务	10分	志愿者活动
寒暑假社会实践	75分	三下乡、企业实践等

核心：早规划、勤积累，别临毕业抓瞎。

第七章： 开放世界 —— 眼光放长远

7.1 支线任务：竞赛与科研

竞赛副本

类型	推荐赛事	特点
算法向	蓝桥杯、ACM/ICPC、力扣周赛	刷题练级，锻炼算法思维
项目向	互联网+、服务外包大赛、软件杯	团队协作，锻炼项目能力

建议：大一打基础，大二开始参赛。竞赛经历对保研/求职都是加分项。

科研初探

- 了解前沿方向：大语言模型、AI智能体、具身智能、软件工程等
- 计算机不只是“修电脑”，还有星辰大海
- 可以尝试联系老师参加大创项目，体验科研

7.2 个人品牌：你的第二张脸

1. GitHub Profile

- 持续上传项目和代码
- 绿色格子（contribution graph）是你的技术简历

2. 技术博客

- CSDN / 掘金 / 知乎 / 个人博客
- 记录学习过程、踩坑经历、技术总结
- 输出是最好的学习方式

3. 社交媒体

- 关注技术大V、参与技术社区讨论
- 拓展视野，获取信息

第八章：🌟 人生航帆 —— 选择你的未来之路

大三之后的分岔路口，没有标准答案，只有适合你的路。

8.1 学术深造

💡 保研（Rank榜冲刺）

- **核心：**这是一场持续三年的长跑
- **关键因素：**GPA + 科研潜力 (Paper) + 竞赛奖项
- **重要节点：**大三下的"夏令营"和"预推免"是关键战役
- **建议：**从大一开始规划，保持年级前列

⚔️ 考研（副本Grind）

- **核心：**选择 > 努力
- **关键：**打破"信息差"比埋头苦读更重要
- **特点：**这是一场孤独的战斗，拼的是心态和自律
- **建议：**确定目标院校，制定详细计划，坚持到底

✈️ 出国（跨服远征）

- **核心：**语言 + 绩点 + 经济支持
- **建议：**尽早考雅思/托福/GRE
- **适合人群：**想体验不同教育体系或有海外发展规划

8.2 职场就业

💼 实习（实战演练）

- **核心：**尽早接触真实工业界代码

- **黄金期**: 大二/大三暑假
- **真相**: 一段大厂实习经历 > 10个课程大作业

面试准备

- **算法**: LeetCode Hot 100 必刷
- **八股文**: 计算机网络、操作系统、数据库基础
- **项目**: 准备2-3个可以深入讲解的项目

技术栈选择

- **T型人才**: 一专多能
- 选定方向 (后端/前端/AI算法/移动端) 深耕
- 用 GitHub 项目证明你的代码能力

8.3 旷野探索

创业 (建公会)

- 需要极强的抗压能力和商业嗅觉
- 适合有改变世界梦想的"疯子"

独立开发 (独行侠)

- 产品力 + 运营能力
- 做自己喜欢的产品，养活自己
- 一人即是一支队伍

Gap Year (游历)

- 人生不是轨道，而是旷野
- 停下来思考，也许比盲目奔跑更有意义

第九章: 存档点 —— 送给大家的话

9.1 三句话寄语

"迷茫是正常的，行动是最好的解药。"
"身体是革命的本钱 (少熬夜，多运动)。"

"保持好奇心 (Stay Hungry, Stay Foolish)。"

9.2 最后的忠告

1. **现在开始，永远不晚** —— 无论之前荒废了多少，从今天开始改变
2. **不要和别人比** —— 每个人的节奏不同，做好自己的规划
3. **多试错，少纠结** —— 大学是成本最低的试错期
4. **建立自己的信息渠道** —— 学长学姐、老师、知乎、GitHub、技术社区
5. **学会求助** —— 不懂就问，没人会笑话你

特别推荐：科研招募

对软件测试、软件缺陷定位技术感兴趣的同学看过来！

如果你想参加大创项目或体验科研，但又担心没有经验、怕老师太严厉……

 **强烈推荐王兴亚老师！**

王老师人超级Nice，不仅学术功底深厚，最重要的是非常有耐心且乐于沟通。无论你的想法是否成熟，他都愿意倾听并给予支持，真的是我们在科研路上的宝藏老师！

联系方式

随时欢迎交流 (Q&A)

- **WeChat:** lhp18260181785
- **GitHub:** [个人主页链接](#)

祝各位期末顺利，四年后我们都能成为更好的自己！