

ACM 二战老油条的经验分享

SiriYang、YY、老龙、海峰“学姐”

2019.5

从大一开始参加 ACM 以来到今年已经是第三次去 ACM，第一次参加打了酱油空手而归，当时再多做一道题就能拿奖，第二次才在两位牛皮队友的配合下混得一个铜奖。这也是我目前参加过难度最高最有含金量的一个比赛，对于这个比赛还是感触良多。在此在队友以及往届学长的帮助下总结过去两年比赛的一些经验、技巧以及一些血淋淋的教训，希望对第一次参加，不是很了解这个比赛的同学有所帮助，参考借鉴。也希望我们的经验总结能流传下去对今后成大的 ACM 发展有一份帮助。

先简要介绍一下这个比赛。本次 ACM 比赛是四川片区比赛，参加比赛的学校除了四川的高等院校以外重庆的高校也划在四川片区。过去的两年金奖基本上是被电子科大、重庆大学、重庆邮电大学垄断，很少有非 985、211 高校能拿金牌，获得金牌也意味着可以晋级更高级的比赛。在去年就有一匹成都信息工程大学的黑马杀入金牌，但也只是最后一名金牌，打了个擦边球，这支队伍之后我还会提到。而我们学校近年来最好的成绩是银奖，但也是好几年以前的事情了。

比赛的具体流程如下：

第一天报道领取队服及相关资料，下榻宾馆。在下午有一个热身赛，届时可以到明天正式比赛的场地机位试座，先体验一下环境。热身赛除了几道题目练手题目以外其余的和正式比赛一模一样。不过不必在热身赛的题目上纠结太多，凡是考到的题目正式比赛都不会考，2333...通过热身赛把你能想到的各种不确定因素如提交代码方法，提交格式之类的问题都测试一遍，确保万无一失，同时也能提前了解一下比较强势的队伍是那些。

第二天上午九点比赛开始一直持续到下午两点，总共五个小时，中午主办方会发干粮。比赛结束之后到指定会场进行颁奖仪式，现场公布排名颁奖，获奖队伍会有一块大奖牌（金牌和铜牌颜色特别像，有时候傻傻分不清）。颁奖结束就可以拍照留念，安排回家了。

接下来是最重要的经验分享环节：

1. 认真读题，当成阅读理解来做

首先重要的事情说三遍，**认真读题、认真读题、认真读题**。一定要认真阅读每一道题的题干，真正弄明白题目要求你要去做什么以后在开始设计算法编写代码。因为我不止一次因为没有完全正确的理解题意就开始盲目解题，导致最后答案错误、所有工作白费，得从头再来。甚至更惨的是到最后都没弄明白题目到底要求我们做什么。如果说混铜奖的关键是作对那几道简单题的话，那此时的 ACM 与其说是在考算法，不如说是在考阅读理解，英语反而成了制胜关键，当然难题更是如此。特别要注意题目中描述的数学用语（如 odd、even 的区别），边界值，特殊值。

2. 读完所有题目

题目发下来以后一定要将所有题目先看完再做，先挑简单的，会做的来做，如果卡住

了就马上跳下一道，不要死磕一道题浪费时间。

3. 如何选择题目

ACM 一共大概 12 到 13 道题左右，出第一道题最简单，剩余的题目难度随机排序，所以你一开始根本不知道后面哪道是简单题那道是难题，有些题你看似简单可能里面有坑。如果不是追求金牌的话最简单的办法就是等大佬先做，一般那些牛队几分钟就能 AC 第一道题，看他们哪道题先做出来，那你就跟着做。到了中后期你就看哪道题 AC 率最高，那这些题就相对简单。那些没人碰的题，或过关率很低的题不建议碰。同时为了防止 AK（及所有题都被做出来），专门会安排一道极难的题，那道题不用考虑，基本上不会有人做的出来，不然组委会就打脸了。

4. 队伍配合

一人负责一题的编码，其余两人只提供思路。如果你想到了好的思路，不管如何都要等当前的队友完成，千万不要打断他，这是 YY 学长血的教训，因此而痛失银牌。我个人的建议是在草稿纸上把详细的算法思路，所使用的数据结构商量清楚了再开始编程，无论是数学建模，还是实际工程开发都是如此，相对于详细的文档设计，编程反而显得次要，要是一开始方向就错了，写再多也是徒劳。

5. 封榜

因为比赛的排名是实时的，但为了增加竞赛的悬念，在最后一个小时排行榜会被锁定，之后一个小时的排名变动选手无法看到，数据会停留在封榜的那一刻。最终排名会在揭晓获奖名单时公布。

但到了最后一个小时还能做出来的一般少之又少，所以变化不会很大，如果不是刚好在铜奖的倒数十名左右的话应该就稳了。因为那些简单题大家能做出来的早就做出来了，剩下的变化基本上都是在高手队伍之间进行较量，他们之间的排名变动不会对你有太大影响，对于越靠后的队伍反而压力没那么大，就算有新的队伍做出了简单题，也会因为提交太晚而排在后面（前提是你别罚时太多）。

6. 准备资料

ACM 竞赛是允许携带任何纸质资料进场的，同学们需要提前准备好英语字典、算法书、代码模板。（对于算法书和模板在下方有所推荐以及资源链接）

7. 使用 Java 语言编写代码

个人推荐使用 Java 语言进行编程。

在允许使用的标准 Java 环境下有丰富的工具类可供使用，省时省力，还能有较高的执行效率以及正确性。再有就是相较于官方提供的编译软件，Java 的 eclipse 要比 C/C++ 的 code blocks 好用的多，不管是调试还是代码提示上（如果你习惯了用 VS 编写 C/C++ 的话可能会很不适应）。

还有一点推荐 Java 的原因在于 Java 的输入输出方法较为丰富完善，对于 ACM 这个很吃数据格式的比赛来说，这是至关重要的一点。在去年的比赛中我们队已开始选择的 C++ 编写，结果本地调试没问题，提交上去就是过不了输入老是有问题，最后一模一样的算法换成 Java 来运行就直接通过了，之后就全程使用的 Java 编写代码，虽然 C++ 肯定也能正确写出来，但确实要麻烦一些。

不同的算法使用 C++ 与 Java 各有优势，C++ 的一些算法实现更为轻便简洁，Java 显得

有点繁重，特别是高手更喜欢用 C++ 来编写算法，但对于基础相对薄弱的新手来说还是老老实实用 Java 吧。

对于 C 和 Java 的速度差异完全不需要考虑，个别题目对于 Java 会单独设置相应的时间空间要求，如果真的遇到超时的问题那肯定是算法上的运行时间和题目要求有数量级的差距，更因该考虑优化算法而不是编程语言的锅。

重点就是要熟练掌握 Java 的各种常用类、各种输入输出函数以及 eclipse 的使用。

8. Java 的提交格式

Java 代码在提交的时候要注意，提交的文件不能有包名，只能 import Java 1.8 的标准下的包，代码文件及主类名要取名为 Main：

```
1. import xxx
2.
3. public class Main(){
4.
5.     public static void main(){
6.         //算法代码
7.     }
8.
9. }
```

9. 使用对数器，尽量少提交

如果是想要混个铜奖其实很简单，你确实只需要把最简单的那几道题做出来就够了，大概也就是三到四道题的样子。而在大家都只做出简单题的情况下还想争夺奖牌的话那就比谁罚时少，提交时间早。而我之前提到的那支成信的黑马当时比金牌的倒数第二名少做了两道题，这中间完全差了两个数量级。但他们之所以能甩掉后面的队伍获得金牌在于他们提交次数较少罚时很少，所以才勉强闯入金牌。

一次提交错误罚时 20 分钟，编译错误不算。但往往只跑通题目给出的数据不能保证我们的答案正确。想要少提交同时在本地验证代码的正确性，此时就要使用对数器。所谓对数器就是你要先写出一个简单暴力但又是完全正确的算法，该算法容易实现但时间复杂度很高，此时再写一个优化算法，使用一个随机函数按照题目要求生成模拟数据，让两个算法同时跑，比较每一条输出结果。只要完全相同就表示优化算法正确，如果错误就打印出来再针对问题数据修改算法。这是我写的一个 Java 代码标准库文档，其中第一个代码就是对数器模板，供大家参考。

<https://github.com/SiriYXR/JSCL/tree/master/doc>

对相关知识点的备战要求：

模拟题：

洛谷上可以搜，大家最需要过的关

数论：

数论是 ACM 的常考知识点之一，对数学功底要求较高，至少需要掌握：辗转相除法

(欧几里得定理)、拓展欧几里得定理、分解质因数、素数的判断、快速幂

广度优先搜索：

和模拟题一样，有点麻烦，比较吃编程能力，也要过关，主要是掌握队列的使用以及判重，广搜的难点在于条件的判定，判定部分实际上相当于模拟题

图论：

最短路算法的几个变种应该熟练掌握

深度优先搜索：

全排列全组合应该做到能背着随手写出来，其它如迷宫问题、八皇后问题也应该有明确的思路

动态规划：

背包问题是最基本的，01 背包完全背包都应该有思路，重点是掌握“状态转移方程”，建议多去 POJ ZOJ HDU 等网站搜几道简单点的动态规划题，反复分析练习，体会状态转移的概念

博弈：

记忆化搜索的特例，也可以看看，至少要能解简单的博弈题

并查集：

原理简单，但是如果有变化，需要有较强的编程能力，在模拟题中锻炼

线段树：

掌握基本原理就行，做到能用线段树解约瑟夫问题

最后推荐一点优质的学习资源：

1. 各大 OJ 网站
实战才是检验真理的唯一标准，在各大 OJ 网站上也能学习别人的优秀解题代码
2. 视频资源
牛客网左程云算法基础班教程，左神华中科技大学本科文凭，芝加哥大学硕士文凭，也是为常年混迹算法的 ACM 大牛，其课程讲解思路清晰明了，对于一些算法思想的总结精炼到位，**强烈推荐大家学习**。
(PS：由于是收费课程，我也是蹭的朋友购买的网课，有需要的同学可以下载百度网盘的盗版资源，不过盗版视频最后几集被设为加密文件无法观看，有点可惜，之后有时间可以考虑为大家重新录制一下，<https://pan.baidu.com/s/1ZzhjxyjZba-0snoxzM54gg>)
3. 书籍资料
挑战程序设计竞赛（第二版），注意该系列还有一本讲数据结构的，个人建议先看看这本，本书是几位日本的国际 ACM 大牛所著，由浅入深的讲解了 ACM 的算法体系。

ACM 国际大学生程序设计竞赛-基础算法（旧版为算法与实现），本系列是上海交大历届 ACM 大佬所著，其中本书记录了大量丰富的 C++ 算法模板，可以在比赛的时候准备一本，其中新版和旧版的区别在于新版按照算法思想分类，旧版按照数据结构分类，两者覆盖范围大致相同均可使用。