# 如何练习ACM

参加ACM比赛所需的基础知识

一、语言是最重要的基本功

无论侧重于什么方面，只要是通过计算机程序去最终实现的竞赛，语言都是大家要

过的第一道关。亚洲赛区的比赛支持的语言包括C/C++与JAVA。笔者首先说说JAVA，众所

周知，作为面向对象的王牌语言，JAVA在大型工程的组织与安全性方面有着自己独特的

优势，但是对于信息学比赛的具体场合，JAVA则显得不那么合适，它对于输入输出流的

操作相比于C++要繁杂很多，更为重要的是JAVA程序的运行速度要比C++慢10倍以上，而

竞赛中对于JAVA程序的运行时限却往往得不到同等比例的放宽，这无疑对算法设计提出

了更高的要求，是相当不利的。其实，笔者并不主张大家在这种场合过多地运用面向对

象的程序设计思维，因为对于小程序来说这不旦需要花费更多的时间去编写代码，也会

降低程序的执行效率。

接着说C和C++。许多现在参加讲座的同学还在上大一，C的基础知识刚刚学完，还没

有接触过C++，其实在赛场上使用纯C的选手还是大有人在的，它们主要是看重了纯C在效

率上的优势，所以这部分同学如果时间有限，并不需要急着去学习新的语言，只要提高

了自己在算法设计上的造诣，纯C一样能发挥巨大的威力。

而C++相对于C，在输入输出流上的封装大大方便了我们的操作，同时降低了出错的

可能性，并且能够很好地实现标准流与文件流的切换，方便了调试的工作。如果有些同

学比较在意这点，可以尝试C和C++的混编，毕竟仅仅学习C++的流操作还是不花什么时间

的。

C++的另一个支持来源于标准模版库（STL），库中提供的对于基本数据结构的统一

接口操作和基本算法的实现可以缩减我们编写代码的长度，这可以节省一些时间。但是

，与此相对的，使用STL要在效率上做出一些牺牲，对于输入规模很大的题目，有时候必

须放弃STL，这意味着我们不能存在“有了STL就可以不去管基本算法的实现”的想法；

另外，熟练和恰当地使用STL必须经过一定时间的积累，准确地了解各种操作的时间复杂

度，切忌对STL中不熟悉的部分滥用，因为这其中蕴涵着许多初学者不易发现的陷阱。

通过以上的分析，我们可以看出仅就信息学竞赛而言，对语言的掌握并不要求十分

全面，但是对于经常用到的部分，必须十分熟练，不允许有半点不清楚的地方，下面我

举个真实的例子来说明这个道理——即使是一点很细微的语言障碍，都有可能酿成错误

：

在去年清华的赛区上，有一个队在做F题的时候使用了cout和printf的混合输出，由

于一个带缓冲一个不带，所以输出一长就混乱了。只是因为当时judge team中负责F题的

人眼睛尖，看出答案没错只是顺序不对（答案有一页多，是所有题目中最长的一个输出

），又看了看程序发现只是输出问题就给了个Presentation error（格式错）。如果审

题的人不是这样而是直接给一个 Wrong Answer，相信这个队是很难查到自己错在什么地

方的。

现在我们转入第二个方面的讨论，基础学科知识的积累。

二、以数学为主的基础知识十分重要

虽然被定性为程序设计竞赛，但是参赛选手所遇到的问题更多的是没有解决问题的

思路，而不是有了思路却死活不能实现，这就是平时积累的基础知识不够。今年World

Final的总冠军是波兰华沙大学，其成员出自于数学系而非计算机系，这就是一个鲜活的

例子。竞赛中对于基础学科的涉及主要集中于数学，此外对于物理、电路等等也可能有

一定应用，但是不多。因此，大一的同学也不必为自己还没学数据结构而感到不知从何

入手提高，把数学捡起来吧！下面我来谈谈在竞赛中应用的数学的主要分支。

1、离散数学——作为计算机学科的基础，离散数学是竞赛中涉及最多的数学分支，

其重中之重又在于图论和组合数学，尤其是图论。

图论之所以运用最多是因为它的变化最多，而且可以轻易地结合基本数据结构和许

多算法的基本思想，较多用到的知识包括连通性判断、DFS和BFS，关节点和关键路径、

欧拉回路、最小生成树、最短路径、二部图匹配和网络流等等。虽然这部分的比重很大

，但是往往也是竞赛中的难题所在，如果有初学者对于这部分的某些具体内容暂时感到

力不从心，也不必着急，可以慢慢积累。

竞赛中设计的组合计数问题大都需要用组合数学来解决，组合数学中的知识相比于

图论要简单一些，很多知识对于小学上过奥校的同学来说已经十分熟悉，但是也有一些

部分需要先对代数结构中的群论有初步了解才能进行学习。组合数学在竞赛中很少以难

题的形式出现，但是如果积累不够，任何一道这方面的题目却都有可能成为难题。

2、数论——以素数判断和同余为模型构造出来的题目往往需要较多的数论知识来解

决，这部分在竞赛中的比重并不大，但只要来上一道，也足以使知识不足的人冥思苦想

上一阵时间。素数判断和同余最常见的是在以密码学为背景的题目中出现，在运用密码

学常识确定大概的过程之后，核心算法往往要涉及数论的内容。

3、计算几何——计算几何相比于其它部分来说是比较独立的，就是说它和其它的知

识点很少有过多的结合，较常用到的部分包括——线段相交的判断、多边形面积的计算

、内点外点的判断、凸包等等。计算几何的题目难度不会很大，但也永远不会成为最弱

的题。

4、线性代数——对线性代数的应用都是围绕矩阵展开的，一些表面上是模拟的题目

往往可以借助于矩阵来找到更好的算法。

5、概率论——竞赛是以黑箱来判卷的，这就是说你几乎不能动使用概率算法的念头

，但这也并不是说概率就没有用。关于这一点，只有通过一定的练习才能体会。

6、初等数学与解析几何——这主要就是中学的知识了，用的不多，但是至少比高等

数学多，我觉得熟悉一下数学手册上的相关内容，至少要知道在哪儿能查到，还是必要

的。

7、高等数学——纯粹运用高等数学来解决的题目我接触的只有一道，但是一些题目

的叙述背景往往需要和这部分有一定联系，掌握得牢固一些总归没有坏处。

以上就是竞赛所涉及的数学领域，可以说范围是相当广的。我认识的许多人去搞信

息学的竞赛就是为了逼着自己多学一点数学，因为数学是一切一切的基础。

回复4:acm程序大赛除了要学好数据结构还需要学好哪些知识？

三、数据结构与算法是真正的核心

虽然数学十分十分重要，但是如果让三个只会数学的人参加比赛，我相信多数情况

下会比三个只会数据结构与算法的人得到更为悲惨的结局。

先说说数据结构。掌握队列、堆栈和图的基本表达与操作是必需的，至于树，我个

人觉得需要建树的问题有但是并不多。（但是树往往是很重要的分析工具）除此之外，

排序和查找并不需要对所有方式都能很熟练的掌握，但你必须保证自己对于各种情况都

有一个在时间复杂度上满足最低要求的解决方案。说到时间复杂度，就又该说说哈希表

了，竞赛时对时间的限制远远多于对空间的限制，这要求大家尽快掌握“以空间换时间

”的原则策略，能用哈希表来存储的数据一定不要到时候再去查找，如果实在不能建哈

希表，再看看能否建二叉查找树等等——这都是争取时间的策略，掌握这些技巧需要大

家对数据结构尤其是算法复杂度有比较全面的理性和感性认识。

接着说说算法。算法中最基本和常用的是搜索，主要是回溯和分支限界法的使用。

这里要说的是，有些初学者在学习这些搜索基本算法是不太注意剪枝，这是十分不可取

的，因为所有搜索的题目给你的测试用例都不会有很大的规模，你往往察觉不出程序运

行的时间问题，但是真正的测试数据一定能过滤出那些没有剪枝的算法。实际上参赛选

手基本上都会使用常用的搜索算法，题目的区分度往往就是建立在诸如剪枝之类的优化

上了。

常用算法中的另一类是以“相似或相同子问题”为核心的，包括递推、递归、贪心

法和动态规划。这其中比较难于掌握的就是动态规划，如何抽象出重复的子问题是很多

题目的难点所在，笔者建议初学者仔细理解图论中一些以动态规划为基本思想所建立起

来的基本算法（比如Floyd-Warshall算法），并且多阅读一些定理的证明，这虽然不能

有什么直接的帮助，但是长期坚持就会对思维很有帮助。

四、团队配合

通过以上的介绍大家也可以看出，信息学竞赛对于知识面覆盖的非常广，想凭一己

之力全部消化这些东西实在是相当困难的，这就要求我们尽可能地发挥团队协作的精神

。同组成员之间的熟练配合和默契的形成需要时间，具体的情况因成员的组成不同而不

同，这里我就不再多说了。

五、练习、练习、再练习

知识的积累固然重要，但是信息学终究不是看出来的，而是练出来的，这是多少前

人最深的一点体会，只有通过具体题目的分析和实践，才能真正掌握数学的使用和算法

的应用，并在不断的练习中增加编程经验和技巧，提高对时间复杂度的感性认识，优化

时间的分配，加强团队的配合。总之，在这里光有纸上谈兵是绝对不行的，必须要通过

实战来锻炼自己。

大家一定要问，我们去哪里找题做，又如何检验程序是否正确呢？这大可不必担心

，现在已经有了很多网上做题的站点，这些站点提供了大量的题库并支持在线判卷，你

只需要把程序源码提交上去，马上就可以知道自己的程序是否正确，运行所使用的时间

以及消耗的内存等等状况。下面我给大家推荐几个站点，笔者不建议大家在所有这些站

点上做题，选择一个就可以了，因为每个站点的题都有一定的难易比例，系统地做一套

题库可以使你对各种难度、各种类型的题都有所认识。

1、Ural：

Ural是中国学生对俄罗斯的Ural州立大学的简称 ，那里设立了一个Ural Online

Problem Set，并且支持Online Judge。Ural的不少题目算法性和趣闻性都很强，得到了

国内广大学生的厚爱。根据“信息学初学者之家”网站的统计，Ural的题目类型大概呈

如下的分布：

题型 搜索 动态规划 贪心 构造 图论 计算几何 纯数学问题 数据结构 其它

所占比例 约10% 约15% 约5% 约5% 约10% 约5% 约20% 约5% 约25%

这和实际比赛中的题型分布也是大体相当的。有兴趣的朋友可以去看看。

2、UVA：

UVA代表西班牙Valladolid大学(University de Valladolid)。该大学有一个那里设

立了一个PROBLEM SET ARCHIVE with ONLINE JUDGE ，并且支持ONLINE JUDGE，形式和U

ral大学的题库类似。不过和Ural不同的是，UVA题目多的多，而且比较杂，而且有些题

目的测试数据比较刁钻。这使得刚到那里做题的朋友往往感觉到无所适从，要么难以找

到合适的题目，要么Wrong Answer了很多次以后仍然不知道错在那里。 如果说做Ural题

目主要是为了训练算法，那么UVA题目可以训练全方位的基本功和一些必要的编程素质。

UVA和许多世界知名大学联合办有同步网上比赛，因此那里强人无数，不过你先要使自己

具有听懂他们在说什么的素质：）

3、ZOJ：

ZOJ是浙江大学建立的ONLINE JUDGE，是中国大学建立的第一个同类站点，也是最好

和人气最高的一个，笔者和许多班里的同学就是在这里练习。ZOJ虽然也定位为一个英文

网站，但是这里的中国学生比较多，因此让人觉得很亲切。这里目前有500多道题目，难

易分配适中，且涵盖了各大洲的题目类型并配有索引，除此之外，ZOJ的JUDGE系统是几

个网站中表现得比较好的一个，很少出现Wrong Answer和Presentation error混淆的情

况。这里每月也办有一次网上比赛，只要是注册的用户都可以参加。

说起中国的ONLINE JUDGE，去年才开始参加ACM竞赛的北京大学现在也建立了自己的

提交系统；而我们学校也是去年开始参加比赛，现在也有可能推出自己的提交系统，如

果能够做成，到时候大家就可以去上面做题了。同类网站的飞速发展标志着有越来越多

的同学有兴趣进入信息学的领域探索，这是一件好事，同时也意味着更激烈的竞争，希

望大家都能通过竞争锻炼自己、提高自己，并争取成为胜利者。

回复5:acm程序大赛除了要学好数据结构还需要学好哪些知识？

ZOJ我前天去过了，就是因为做不出那里的题，所以向大家请教需要哪些知识？在这里我希望大家能够建议我几本比较好的算法书。

回复6:acm程序大赛除了要学好数据结构还需要学好哪些知识？

知识并不是最重要的

快速的解题能力才是这个比赛的关键

其实里面要求的知识大多很初等，一些论文题除外