

OI 比赛

所有的如蓝桥杯、河北大学生程序设计大赛等比赛统称算法竞赛，简称 OI，所以如果看到如所谓 Oler、acmer、icpcer，就是在指自己是算法竞赛选手。算法竞赛一般来说就是给你一段时间和一些程序题，通过写出**正确**同时**高效**的程序来解决其对应的问题。在咱们学校的话能够接触到的比赛基本为学校校赛、蓝桥杯、河北大学生程序设计大赛（简称hbcpc，后面都用这个简称）、天梯赛（没打过，不确定）、icpc。

赛制

比赛的本质是写程序回答问题，但不同的比赛间比赛规则差异很大。一般来说有两大主类，分别称为 **noip赛制** 和 **acm赛制**。

noip赛制（包含noip csp noi 蓝桥杯）

- 个人赛。
- 禁止携带任何资料。
- 没有实时反馈。也就是说交上去的题你是不能够知道你写的是不是对的，所以也没有实时排名一说。
- 有部分分。一道满分100的题你可以拿到30分或70分。
- 可以带食物进入考场~

acm赛制（包含hbcpc icpc acm）

- 团队赛，三人一队。
- 可以携带纸质资料。
- 有实时提交和实时排名。也就是说你的题目交上去之后便可以知道你写的对不对，错误答案会罚时。不对可以重复提交。
- 没有部分分。每道题你只能0分或满分。
- 可以带食物进入考场~

校赛和其它

- 校赛是个人赛，不能携带资料。其它与**acm赛制**一致。
- 天梯赛不了解，问打过的学长吧。

难度（个人参考）

noip、noi、csp一般是给高中竞赛生打的，所以不考虑了。

校赛 < 蓝桥杯省赛 < hbcpc < 蓝桥杯国赛 < icpc \approx acm

获得蓝桥杯省三及以上的奖项就可以过**六个一**了，icpc 或 acm 获奖会有**保研加分**。

参赛指南的话，我强烈建议按照难度进行参赛。

备赛

放在所有话之前的是警告。算法竞赛本身的入门门槛很高，在打到 icpc、acm 阶段时，已经和如数学竞赛一样属于奥赛范畴。因此在进行准备时，我强烈建议结合自身的水平和时间，去为了达到**你能够达到的目标**进行合理的规划，**拒绝盲目自信**。我分成了 3 类目标：

1. 为了六个一，拿到蓝桥杯省三就算胜利。

2. 感觉自己有实力，尝试冲刺 hbcpc 与蓝桥杯国赛。
3. 感觉自己有天赋，目标是 icpcer 或 acmer 。

PS：基本素养和练习方法两块是通用的。

基本素养

- 有一门熟练的编程语言，个人建议 C++。因为 C++ 和 C 语法兼容，且相比于 C 内置了许多 STL，可以方便竞赛时直接调用。
- 对算法、算法复杂度有一个基本的概念。（它们是什么，在比赛中有什么作用）
- 有高中数学基础。
- 学会善用搜索软件，大部分你需要的知识都可以百度出来。
- **有刷题的耐心。**（后面所提到的刷题都是指在刷题网站上写代码，不是传统意义上的写卷子或者只学思路但不写）

目标1：蓝桥杯省三

这个目标是很轻松的，如果你狠的下心来，在有基本素养的前提下**只刷题**1个月左右就可以轻松蓝桥杯省三。

此时的你面对任何题目只要干一件事：完全不管复杂度和数据量问题，你只要把题的要求**模拟**出来。（**模拟**指一种题目要求怎么做，你写的代码就完全能按照要求对应上的一种能力）

- 了解一下一般题目的套路，这里指要学会怎么接收数据和输出答案。
- 对变量、判断、循环、输入输出语句能够熟练运用。
- 时间复杂度是你完全不需要担心的，现在的你只要关心你写的是否正确，无论效率多差都没关系。
- 了解**递归**的原理，争取能够写的出来，毕竟**递归**也是一种重要的**模拟**方式。
- 了解一些基本的数据结构和算法，会**堆栈、队列、排序（指快排）**就够了。

这个阶段最重要的是一种代码思维的培养，需要把重点放在**简单题的练习**上，**数量越多越好**。

当你学会这些之后，你应该去做做蓝桥杯真题。当然你会发现它们中的一部分对你来说非常困难，你要知道这是正常的，因为那是给省一选手做的。OI 比赛与你所见过的许多比赛都不同，这是一门选手水平差距极大的比赛，150分满分的比赛，超过70分的就已经没多少人了，不过同样也会有150分 AK 的大犇。蓝桥杯比赛的题目难度其实中规中矩，但参赛的选手偏水。对于你的目标，你只需要拿到大概20-30分左右，这个分数已经是省三水平了。

目标2：hbcpc、蓝桥杯国赛

从我的经验来看，能完成目标1的人已经是少数了。因此建议拿到一次蓝桥杯省三后再考虑是不是继续向目标2努力。想要进军目标2要从两方面提升自己，一是算法的积累，二是题感与思维能力。

- **找两个志同道合的人组队。**这很重要，自己学会丧失动力。
- 什么都学，做好笔记，多在题解中找灵感，可以考虑建个博客。这个阶段最重要的是知识的积累。
- 保持题感，至少3天1道中等题。
- 关注时空复杂度，搞清楚数据量大小与其对应的极限复杂度，能够准确判断出自己的程序能够拿到多少部分分，少骗分，争取**正解**。
- 熟练掌握**递归和其各种变种用法（记忆化搜索、bfs、dfs）**，这会是你的骗分神器。

数论、图树论、数据结构是你避不开的，虽然它们很难，但不肯钻研这些等于自我放弃。这个阶段可以学的东西太多...我找了一个表，是我当年整理的自我算法掌握情况（因此不保证里面没有错误），可以作为你的参考，里面涉及的知识点上网搜索很容易搜到详解与例题。这张图的知识点是不全的，但即使只掌握了黑字内容也绝对超出了目标2的水平。蓝色为我不熟的内容（或者叫比较难的内容，仅代表几年前的我的水平）。内容很多，但其中有 1/3 左右也是大学《离散数学》与《数据结构》等的内容，因此学习它们也算是提前学习了学科内容。

OI

字符串

- KMP
- 后缀数组 倍增
- 字符串Hash
- Trie树
- manacher (马拉车)

计算几何

- 基础知识
 - %理论
 - 积性函数
 - 高中部分数学知识
- 素数
 - 暴力判素数
 - 筛法求素数
 - 埃拉托色尼筛
 - 线性筛 (欧拉筛)
 - 分解质因数
- 欧拉函数
 - $\text{sqrt}(n)$ 求单个值的欧拉函数
 - 线性筛欧拉函数
- 快速幂
 - 欧拉定理
- gcd
 - 辗转相除法
 - gcd
 - lcm
 - exgcd
 - 求逆元
 - 求同余方程
 - 求 $ax+by=c$
- 中国剩余定理 (CRT)
 - 互质版
 - 不互质版

数论

- 矩阵
 - 矩阵快速幂
 - 矩阵求逆
- 莫比乌斯反演 莫比乌斯函数
- 狄利克雷卷积
- 容斥定理
- 费马小定理
 - 线性求逆元
- 逆元
 - exgcd求逆元
 - 费马小定理求逆元
- 高斯消元
- 线性基
- 排列组合
 - 杨辉三角
 - Lucas定理
- 数列
 - 斐波那契数列 $\text{gcd}(f(n), f(m)) = f(\text{gcd}(n, m))$
 - 等差、等比数列
 - 通项公式
 - 求和公式
- po? ly定理
- 快速傅里叶变换
- 线性规划
- 单调性
- 概率和期望

动态规划

- 简单dp
 - 01背包
- 背包dp
 - 完全背包
 - 多重背包
 - 分组背包
- 区间dp
- 状压dp
- 树形dp
- 环形dp
- 期望dp
- 其他好玩的dp
- 多维动态规划
- dp套dp
- dp的优化
 - 数据结构优化dp
 - 单调队列优化dp
 - 线段树优化dp
 - 寻址优化
- 二分
 - 三分
 - 整体二分
 - 二分答案
- 倍增
- 贪心
- 枚举
- 暴力
- 分治
- 离散化
- 模拟
- meet in the middle

博弈论

- SPFA判负环 SPFA
- 堆优化的dijkstra dijkstra
- 线段树优化的dijkstra
- 最长路
- 差分约束
- 强连通分量/缩点
- 双连通分量 (点、边)
- Tarjan
- 割边、割点、桥

图论

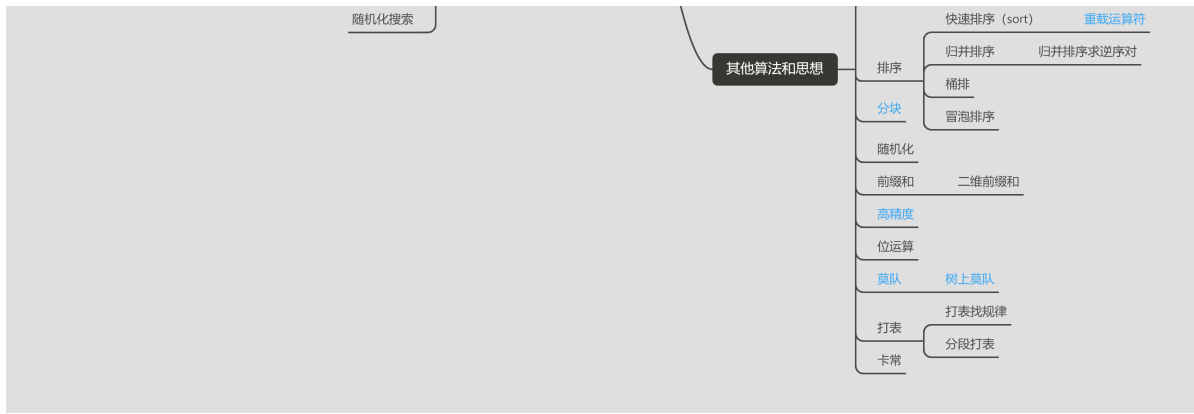
- 二分图
- 网络流
- 欧拉图
- 生成树
 - Kruskal
 - Prim及其堆优化
 - (严格)次小生成树
 - 其他各种生成树
 - 生成树计数
 - 暴力统计 (划掉)
 - Matrix-tree定理
- LCA
 - 倍增
 - 树剖
 - tarjan
- 虚树
- 树链剖分
- dfs序
- 树的dfn?
- 树归?
- 树的遍历
 - 前序
 - 中序
 - 后序
- 树上倍增
- 树的直径
- 树的重心
- 树分治
 - 点分治
 - 边分治
- 树分块
- 仙人掌
 - 动态仙人掌
 - 仙人掌

数据结构

- 单调队列
- 队列
 - 优先队列 (堆)
 - 双端队列
- 栈
 - 单调栈
- 堆
 - 左偏树
 - 斜堆
 - 可并堆
 - 斐波那契堆
- 并查集
 - 可持久化并查集
 - 带权并查集
- hash表
 - 占坑法?
 - 链表法?
- 链表
- st表
- 树状数组
 - 多维树状数组
 - 树状数组求逆序对
 - 超级树状数组?
- 线段树 (.....)
- 平衡树 (.....)
- 树套树 (.....)
- STL (.....)

搜索

- bfs
 - 双向bfs
- dfs
 - 记忆化搜索
- 剪枝
 - 可行性剪枝
 - 最优性剪枝
 - 玄学剪枝
- 模拟退火
- 遗传算法
- 爬山算法



目标3：icpcer、acmer

目标3与目标2之间的差距仍非常大（主要是一种系统的竞赛思维的培养，外加一些十分华丽的算法的掌握）...你要明白你是要和全国的选手进行比拼（包括所有 211 985 名校的选手）。如果你有目标2的实力和进军目标3的决心，就来找我吧。

练习方法

学竞赛最棒的三件事：刷题、打比赛、~~永~~写博客。

刷题网站：

[洛谷](#)

风格很轻快的刷题网站，个人认为是圈子文化最浓的地方。有优质题解。

[一个洛谷题单博客](#)

如果你难以在洛谷里规划好的刷题顺序，可以试试跟着这个写。

[OpenJudge](#)

很经典的刷题网站。

比赛网站：

[codeforces](#)

首推。是一个俄罗斯网站，英文题面，题目质量都很高，偏向考察思维能力。

[牛客网](#)

比赛挺多挺杂的，有一部分要收费参赛，同样有一部分比赛获奖对你以后找工作有帮助。

[还是洛谷](#)

如果你时间充裕，可以打一打洛谷的月赛。

写博客大家都会，每场比赛后最好都要写一个整理博客。

一些经验与题外话

时间规划

OI 比赛都是拉锯战，时间都在3小时以上。刚参赛的选手一般不能比赛开始高强度思考到比赛结束。我的建议是用第1个小时入状态，把最有把握最有思路的题放在第1个小时后突击。最后的半个小时到1个小时基本是垃圾时间，如果没有手感火热的题其实意义就不大了。

蓝桥杯的填空题

有的简单，但有的是真毒瘤。我不建议总时间在上面花费超过1个小时，因为它们的分值真的很低。举个例子，我两个填空都没写，是省一。

蓝桥杯 ≈ 骗分杯

这个比赛是 noip 赛制，那就离不开**骗分**的核心，一定不要为了想正解去发呆，一定要**多写**，即使是写1个小时的程序只能拿到1分也要写！能骗1分也是分！这件事听上去简单，但我自己就多次掉入这个坑出不来，因为这种感觉就像写数学大题思路卡在一个最关键的节点出不来一样，但你要告诉自己解不了就是解不了，一定要及时止损。

个人模板

建议整理一个自己手写的代码模板库，以便写题的时候及时查询，也可以为比赛时候打印资料提供方便。

icpc 与 acm 的英文题面

这两个比赛是英文题面。建议队伍的英语水平要高，现场查牛津是很痛苦的。

博客

有一个自己的博客是很快乐而且收益巨大的。开始你可以选择在洛谷上建、或者博客园、csdn 上建。当你有了写博客的好习惯，你就可以开始着手于利用框架搭建一个自己的博客了，这不仅是对竞赛有好处，对于你的计算机生涯也是一笔财富。

顺手推销下我的博客：[CYHei_mu的博客](#)。里面还是有一些 OI 相关的内容的。

打比赛 = 罚坐5小时的问题

这个很常见，我打 icpc 也经常罚坐5小时写不出来一道题。OI 生涯中要经历最多的就是失败，0 分退场的比赛经历个三五场是非常正常的事情。如果你打算学到底了，就要学会接受这些并习以为常。