
tricks in OI

test [hyperlink](#)

table of content

- 讲解 NOI 赛制中可以利用的资源.
- 介绍辅助解题的技巧/技术/方法论(?).
- 帮你找到一些隐藏学习资源.
- "我是如何在BJOI2019爆0的"经验分享会.

你在这台装了 NOI linux 的机子上可以利用的资源

- python(version=2.7.x) 5 分钟 python 语法串讲 + 案例编写 (数据生成技巧, 搞个对拍)
 - 无需声明, 首次使用即为定义. 有作用域规则, 用缩进表示代码块层级. 可以不用分号
 - 控制结构会用: 标明. 不要混用 tab 和 space(python 用 elif 而不是 else if).
 - * `with open('1.in', 'w') as fin: print(10, file=fin)`
 - * `print`的几个参数, `file`, `end`, `sep`
 - * 可以用上你们熟悉的"`the length of (%s) is %d`"%('Hello World', len('Hello World'))
 - 在 py3 中是使用 `format` 函数配合 `{}` 占位符; 或者使用 `f-string`
 - `abs = lambda x: x if x >= 0 else -x`
 - 四种数据结构 `list`, `tuple`, `dict`, `set`
 - * `list(x)`; `[1, 2, 3, 5, 8]` **允许不同类型的东西放在一起**
 - `append`, `sort`, `count`
 - * `(a, b, c)`; `(a,)`
 - `l, r = r, l`
 - `(max_flow, flow_cost) = solve_MCMF(graph)`
 - * `dict(a=x, b=y, c=z)`; `{'a': x, 'b': y, 'c': z}` 注意字典的索引必须是 hashable 的, 比如 `int`, `string`
 - `for (k, v) in dictionary.items(): code`
 - `dict.get(key) -> None/value`
 - `dict[val] = qwq`
 - * `set([1, 1, 2, 3, 5, 5, 8, 1])`; `{1, 1, 2, 3, 5, 5, 8, 1}`
 - `a in set_of_odd_numss`

- `for i in set_of_even`
- `intersection, union, issubset, issuperset`

- gdb 简易介绍 (NOI linux 上面版本太低...不太好)

- 使用 `g++ -g` 参数编译, 插入调试信息 (使用 `-g` 参数你的程序会跑得很慢)
- `gdb ./a.out` 载入程序, 或者直接进入 `gdb`, 用 `file` 命令加载
- `gdb` 会显示即将执行的下一行代码.
- 使用 `l` 命令查看源码. 从头开始. 你也可以 `l 1926` 从 1926 行开始看
- `b 13` 在 13 行加入断点
- `r` 从头开始执行程序, 到断点
- `c` 继续执行程序, 直到断点.
- `p val` 打印变量 (它会识别变量类型, 打印地址和值的...只是 NOI linux 上面版本老不好用...)
- `n, s` 向下执行, 是否跟踪函数调用.
- `q` 命令退出
- 你的程序可能会 `signal SIGSEGV` 之类的被 kill (评测会 RE), 在出现这个信号的前面检查数组访问越界, use-after-free (比如迭代器失效)
 - * `vec[x]=insert(vec, 3)` 这个就会有问题, 因为 `c++` 没有规定怎么求值, 它可能是把 `vec[x]` 的地址确定了再进入右侧插入, 然后你的 `std::vector<int>` 因为插入扩充而移动了, 然后就 use-after-free 了.
 - * 嵌套模板需要加入空格, 当然 `c++14` (或许是 `c++11`) 标准干掉了这个问题...`--std=c++11` 参数告诉 `g++` 用什么标准.
- 当然一个麻烦的事情是, 它会把输入输出错误, 以及程序源码混在一个地方显示...(最新的 GDB 已经解决这个问题, 当然你们还是需借助 IDE anjuta)

- 使用 `ulimit -s unlimited` 开栈 (NOI, CTSC, APIO, BJOI, NOIp 都是不用担心评测卡栈的, 当然某些恶心人的 hnoi 并没有这么良心)
- vim 如何使用无插件的 vim-基础配置与操作
- arbiter 如何知道自己的程序有没有爆 stack? 是不是 MLE? 本机评测原则了解一下.

熟悉几个命令 (讲解实际使用案例, 如何用 CLI apps 辅助编码/调试/备份)

```
1 cd <path>
2 ls <path>
3 ls -lah <path>
4 rm (-r, -f)
5 touch
6 cp/mv
7
8 cat 1.in | ./brute_force > 1.out
9 vimdiff 1.ans 1.out
10 cat 1.out | grep ERROR --ignore-case
```

```
11
12 ulimit -s unlimited
13
14 man
15 date
16 xdg-open
17 sudo
18 top/htop(在程序最后加上while(1){}使用ctrl+z挂起,用top/htop查看内存使用情况)
```

```
1 Ctrl+C(break)
2 Ctrl+L(并非真的clear...)
3 Ctrl+D(send EOF to stdin)
4 Ctrl+Z(挂起,配合fg,bg)
```

解题技巧

- 猜性质/打表找规律 (python 比 cpp 更适合这个)
- 按照类型猜解法 (**留好你的确保正确性的暴力**)
 - **这玩意,是不是就是个背包啊! a b**
 - **你会写就写出来,别质疑自己的实力**
 - 是个优化模拟?(类似什么数据结构的流程? 是不是涉及经典问题比如 RMQ)
 - 是个数论?(相信这句话,最难就是 exgcd; 如果出难了就放弃吧.... 练习推导套路和实现技巧是要花费巨量时间的)
 - 是组合计数?(能不能直接写出答案表达式[link](#)? 能不能建立递推式? 能不能容斥 (更进一步地,参考我挂在 github 的文章进行学习) 终极武器是生成函数)
 - 是数据结构?(各凭本事.... 你熟悉这模型就秒了...)
 - * 基础数据结构必须熟悉 (随便拉一个 HLD+SGT 的你能不能秒掉?[link](#)), 经典的利用数据结构方便解决的问题模型要清楚 (区间数颜色, 偏序, 01trie 解决 xor 最值, 线段树合并/dfs 序差分解决子树查询, 平衡树/块状链表解决序列动态结构)
 - * 几个基础想法必须熟悉 (均摊复杂度, 离线/反向, 可持久化数据结构, 分治, 分块/按大小分类)
 - * 如果你会 advanced DS 并且确信能拿到好分数, 别犹豫, 写出来! 有些东西是你没见过就做不了的 (参考 18 年 d2t3)
 - * 怎么样写暴力过得点多?(**不开-O2 优化时, 慎用 STL**, 数组能开 1000 不开 500)
 - * 空间复杂度和时间复杂度的估计与证明 (估计到不了这个难度....)
 - 是图论建模?(优先想 SSSP 和 MST, 对于有向图相关的, 有 tarjan's SCC, toposort 可以用. 常见的问题要知道: 独立集, 团, 点覆盖)
 - 是特化经典算法?
 - * 手动模拟一下, 看看什么地方是同质化的 (例子[link](#))

-
- 用数据结构优化用数据结构优化
 - 是个最优化问题?
 - * 不是 dp 就是二分 check. 总不能是线性规划建图吧
 - * 可以非常难, 参考 19 年 d1t3 和 d2t2
 - * 当然也可以很水参考 18 年 d1t3
 - 怎么写“测试点分治”
 - 善用 namespace
 - * 空间消耗相加, 记得测测内存.
 - 自己测测是不是多组数据会挂

你的训练资源在哪里?

- oier 们的 blog 是最方便, 最常见的资源.
- 一些远古资源我可以发给你们啊 ([呲牙笑])
- OI-wiki 和 codeforces 是你将来获取难以理解的技巧的地方.
- 集训队论文 (尤其是最近 5 年来质量都是非常高; 之前的所谓论文颇有文献综述的味道) 就算你进不了集训队也可以读, 也应该读.
- APIO/CTSC/WC 都有讲课分享环节 (也许叫营员交流), 如果有用一定拿来读一读, 有问题的地方可以直接在uoj group 里面问讲课者本人 (亲身经历, 学 SAM 问了在 APIO 上分享广义 SAM 玩法的 zyf 一堆推导)
- 网友比学长学姐水平更高, 他们甚至比你的教练更有空帮你, 主动融入 OI 的圈子.
 - 推荐阅读
 - * <https://loj.ac/article/2374>
 - * <https://loj.ac/article/2476>
 - * <https://loj.ac/article/1757>

最近题目复盘

2015

- 幻方简单模拟
- 信息传递简单有向基环树找环
- 斗地主 (出题人 sb)
- 跳石头经典中的经典...
- 子串经典的子序列 dp 模型.... 需要一点优化
- 运输计划答案具有单调性的可二分最优化问题 + 经典数据结构

2016

- 组合数问题啊这...不会的话说明高中数学不及格
- 愤怒的小鸟看数据范围就是状态压缩 dp, 熟悉图论优化问题的话发现是经典的最小点覆盖问题
- 蚯蚓你会 STL 吗? 那你有 75pts 了...是个好题目, 不是太难
- 玩具谜题数据结构优化的模拟
- 天天爱跑步经典数据结构模型 (子树统计 \leftarrow 线段树合并/dfs 序差分)+ 形式化问题 (写出那个关于深度的统计式子)
- 换教室这也是个经典问题...(图上随机游走也是类似的经典问题)

2017

- xiaokai 的疑惑数学题都不难, 最多 exgcd
- 时间复杂度怎么写不难, 难在怎么造数据验证自己的程序
 - 这个是手动实现的有限状态自动机 (DFA)
 - 推荐[这个题目](#)
- 逛公园是经典问题, 数据的 $k \leq 50$ 是解题关键
- 奶酪签到
- 宝藏神题, 但是靠直觉可以做, 状态压缩经典题目 (这个当然是这类题目中非常难的一个题了), 你知道这个题的数据很难构造的情况下如何乱搞.
- 列队数据结构, 不会块链/平衡树没救

2018

- 铺设道路签到
- 货币系统这是不是个背包? 就算你和我一样蠢也可以 80pts
- 赛道修建最小化最大值? 二分; 按照大小顺序合并? 数据结构 (或者你知道 `std::map, std::multiset`)
- 旅行送了 60...别想太多你就 AC 了系列
- 填数游戏现场用 python 写个搜索跑一个小时你就有 65 了 (我写的没有任何剪枝都半小时跑出来所有点了...)
- 保卫王国送了经典题的分数, 一条链也是经典题 (但是你要会用数据结构维护矩阵乘法, 如果你做过这种东西就随手拿到了). 更进一步非常难

2019

- 格雷码如何卡自己系列
- 树上的数难度起飞, 我是 10pts 滚蛋
- 括号树序列上是经典问题, 别想太多系列
- emiya 家的饭暴力 $64, a > \lceil \frac{k}{2} \rceil$ 是不是 $2a > k$ 的意思? 容斥
- 划分决策单调性 dp...或者你猜个结论可以秒
- 树的中心如何数据分治系列, 其实也是个经典数据结构题...

杂题选讲

主要是 NOI online 的题目,
codeforces 的题目,

趋势分析

- 签到还是有的, 但是变少了.
- day2 难度起飞是 [noi2018](#) 以后所有 ccf noi 系列比赛的共性.
- 经典题目必须会, 绝不能有漏洞.
- 你会模型就秒杀的题目变少, 有些题目难度起飞了.(当然也引入了新套路...说白了就是省选考烂了就放 noip...)
- 代码量仍然不大, 如果你随手写数据结构题, 那你肯定不会遇到写不完的情况.
- 数学题都出的不难, 别想歪了.
- dp 还是重中之重, 建议所有人都看 LRJ 紫色书的 dp 经典题.
 - 你要会一维 dp 优化
 - 数据结构优化 (单调队列是最简单最常见的)
 - 决策单调性 (这也能 noip 我是想不到的...)
 - 玄妙的分治/改成网络流等神秘 trick 最好也会.
- 你还是得会贪...
- **!!! 你必须会快速实现搜索!!!**
 - 如果你不会, 找省选题来, 读懂题意直接就写, 写上一周就都会了.
- 应当适当地学习一些高级知识与技巧, 以获得启发
 - 做题一定从最直觉或者最简单的方法开始考虑, 不要直接就确定是某个高级 DSA.
 - 如果要学, 不能只学板子.

当然你水平上去了, 随便打也都能在 zj 拿 1=

近来,OI 和其他学科竞赛,遇到了不少风波. 有的来自教育部, 有的来自 CCF.

你可能会觉得自己未来暗淡. 或者十分失望. 自嘲一句“我没学上了”

但是请你记住, 不论是 NOI 还是高考都不是你人生的结束.

你可能被他们视为 loser, 但是你要明白, 中国人的平均寿命已经超过 70 岁了, 就算用 20 年才能建立热爱, 学习基础, 走向前沿, 做出成果. 你也还有三次机会!

OI 不是一个选拔每年 4 人参加 IOI 的体育竞技. 是给热爱 CS 及其相关领域的孩子们一个深入学习知识, 了解领域研究境况和实际应用场景准备的独特学习经历. 从来没有人否定你的兴趣, 否定你的努力, 否定你的进步, 否定你的前景.

当你在今后学 OI 学习与训练, 文化课学习与应试, 高考志愿填报选校中陷入困境时, 请记住你们还有一群活着的学长学姐, 我们会尽所能支持你, 不要犹豫, 给我打个电话, 发给微信消息吧.

最后我想说, 大多数人是普通人, 不论你是在四中, 将来在清华, 后来在 MIT. 你们从来不欠学校什么成绩, 不要给自己加压太多.

平常人的生活, 就是挺好的了, 这个社会, 整个国家和民族乃至全世界的人类群体, 都是需要我们这样的螺丝钉的.