XJOI 进阶讲义（一）

Mr_Spade

2021.9

本节课的内容：对目前的竞赛情况进行一些总结，并针对接下来应该采取的学习方法提出我的建议。最后，交流课程会采取的上课方式。

如果时间比较宽裕，则将继续讲解对 NOI2021 的比赛解析。

- 高二

- 高二
- 一种勇于面对充满挑战的生活的品质

- 高二
- 一种勇于面对充满挑战的生活的品质
- 我的经历

- 高二
- 一种勇于面对充满挑战的生活的品质
- 我的经历
- 目前的时间节点

清北强基优惠

- 门槛降低
- 采取正确方法提升自己的实力

冲击省队、国家集训队

- 不要因为认为自己实力不足而不敢有这样的愿望

冲击省队、国家集训队

- 不要因为认为自己实力不足而不敢有这样的愿望
- OI 从来不是只为顶尖高手准备的游戏

冲击省队、国家集训队

- 不要因为认为自己实力不足而不敢有这样的愿望
- OI 从来不是只为顶尖高手准备的游戏
- 打好基础，稳步前进，同样可以冲击优秀的成绩

冲击省队、国家集训队

- 不要因为认为自己实力不足而不敢有这样的愿望
- OI 从来不是只为顶尖高手准备的游戏
- 打好基础，稳步前进，同样可以冲击优秀的成绩
- 联合省选风格稳定

风险诚存在

- 接受不好的结果发生的可能性

风险诚存在

- 接受不好的结果发生的可能性
- 我们无法预知未来，只能在当下全力以赴

风险诚存在

- 接受不好的结果发生的可能性
- 我们无法预知未来，只能在当下全力以赴
- 我更享受在赛场上的感觉

风险诚存在

- 接受不好的结果发生的可能性
- 我们无法预知未来，只能在当下全力以赴
- 我更享受在赛场上的感觉
- 即使失败，这段时光一定会是你们难忘的回忆

在提升自己能力的路上，我们不能任凭自己的兴趣随性学习，而更要适当忤逆自己的兴趣，学习一些我们相对不那么感兴趣但在比赛中十分重要的知识点。

在提升自己能力的路上，我们不能任凭自己的兴趣随性学习，而更要适当忤逆自己的兴趣，学习一些我们相对不那么感兴趣但在比赛中十分重要的知识点。

对某些方面有着深入研究的兴趣当然是一件好事，但是现在我们更应该将精力投入到竞赛中，在竞赛最终落下帷幕，我们也走上新的道路后，重拾自己的兴趣就是一个很好的选择，而不急于在现在一时就展开研究。

在提升自己能力的路上，我们不能任凭自己的兴趣随性学习，而更要适当忤逆自己的兴趣，学习一些我们相对不那么感兴趣但在比赛中十分重要的知识点。

对某些方面有着深入研究的兴趣当然是一件好事，但是现在我们更应该将精力投入到竞赛中，在竞赛最终落下帷幕，我们也走上新的道路后，重拾自己的兴趣就是一个很好的选择，而不急于在现在一时就展开研究。

我们有时经常会不自觉地忽视这一点，盲目地跟随大流。

想要成为一名优秀的算法竞赛选手，有时需要一种不拘小节的大局观。

想要成为一名优秀的算法竞赛选手，有时需要一种不拘小节的大局观。

我们不仅要注重不需要的知识少学，也要注意重要的知识全面学。

你是否曾有这样的困惑，有些选手明明平时成绩不错，为什么在真正考试时经常给人“靠不住”的感觉，总是掉链子？

而有些选手平时中规中矩，表现一般，考试却总是发挥稳定，甚至取得优秀的成绩？

你是否曾有这样的困惑，有些选手明明平时成绩不错，为什么在真正考试时经常给人“靠不住”的感觉，总是掉链子？

而有些选手平时中规中矩，表现一般，考试却总是发挥稳定，甚至取得优秀的成绩？

比赛素养！

心态调整

- 正式比赛的压力

心态调整

- 正式比赛的压力
- 平时比赛随意，正式比赛就可能碰到未知的情况

心态调整

- 正式比赛的压力
- 平时比赛随意，正式比赛就可能碰到未知的情况
- 做好充足的心理准备

心态调整

- 正式比赛的压力
- 平时比赛随意，正式比赛就可能碰到未知的情况
- 做好充足的心理准备
- 不妨随意一点

有效地拿部分分的能力

我们打比赛不是为了完全通过尽可能多的题，而是拿尽可能高的总分。

有效地拿部分分的能力

我们打比赛不是为了完全通过尽可能多的题，而是拿尽可能高的总分。

我知道愿意在竞赛上一路坚持的同学们或多或少都注重从比赛中获得的荣誉感和成就感。

但是我们更应该想到，与这样的成就感相比，获得更高的分数，站上更高的领奖台，实现我们的梦想，是比获得小小的成就感更让我们收获至高无上的感动和理想的事情。为了实现这份理想，我们连付出自己的青春都在所不惜，在比赛中适当地放弃一些完整做出一道题的快感又何妨呢？

有效地拿部分分的能力

我们打比赛不是为了完全通过尽可能多的题，而是拿尽可能高的总分。

我知道愿意在竞赛上一路坚持的同学们或多或少都注重从比赛中获得的荣誉感和成就感。

但是我们更应该想到，与这样的成就感相比，获得更高的分数，站上更高的领奖台，实现我们的梦想，是比获得小小的成就感更让我们收获至高无上的感动和理想的事情。为了实现这份理想，我们连付出自己的青春都在所不惜，在比赛中适当地放弃一些完整做出一道题的快感又何妨呢？

我始终认为，将模拟赛真正地看作一场比赛是最好的利用这场比赛的方式，你需要从中学会如何在开局利用暴力分打好下盘，如何研究部分分的性质，根据题目的数据范围提示你想到相关的解法甚至引导你的解题思路，也要学会如何在时间有限的前提下取舍部分分来获得尽可能高的收益。

培养获得正解的能力

获取正解的过程其实就和获取暴力分部分分的过程类似，就是一步步深入思考罢了，只不过是因水平不同，最终获得的部分分也不同，如果水平够高，那满分就是我们的“部分分”。

培养获得正解的能力

获取正解的过程其实就和获取暴力分部分分的过程类似，就是一步步深入思考罢了，只不过是因水平不同，最终获得的部分分也不同，如果水平够高，那满分就是我们的“部分分”。

我们可以在一场模拟赛结束后，对着在赛中没想出正解的题目花费更多的时间时间去思考，甚至可以花费比比赛长的多的时间，去想清楚正解中的每一个细节，确认自己的思路明白无误，在最终完美解决这个问题。

培养获得正解的能力

获取正解的过程其实就和获取暴力分部分分的过程类似，就是一步步深入思考罢了，只不过是因水平不同，最终获得的部分分也不同，如果水平够高，那满分就是我们的“部分分”。

我们可以在一场模拟赛结束后，对着在赛中没想出正解的题目花费更多的时间时间去思考，甚至可以花费比比赛长的多的时间，去想清楚正解中的每一个细节，确认自己的思路明白无误，在最终完美解决这个问题。

但是，有一点请务必要注意：不要因为通过了一道题就满意地到此为止，现在通过了这道题，只是你经过了比比赛时间长得多的时间的结果。

培养获得正解的能力

获取正解的过程其实就和获取暴力分部分分的过程类似，就是一步步深入思考罢了，只不过是因水平不同，最终获得的部分分也不同，如果水平够高，那满分就是我们的“部分分”。

我们可以在一场模拟赛结束后，对着在赛中没想出正解的题目花费更多的时间时间去思考，甚至可以花费比比赛长的多的时间，去想清楚正解中的每一个细节，确认自己的思路明白无误，在最终完美解决这个问题。

但是，有一点请务必要注意：不要因为通过了一道题就满意地到此为止，现在通过了这道题，只是你经过了比比赛时间长得多的时间的结果。

关于题解以及证明

选择合适的高质量题目作为训练用题

- 题目丰富但鱼龙混杂

选择合适的高质量题目作为训练用题

- 题目丰富但鱼龙混杂
- 题目的缺点：算法生僻、部分分不充足

选择合适的高质量题目作为训练用题

- 题目丰富但鱼龙混杂
- 题目的缺点：算法生僻、部分分不充足
- 优秀题目来源：真题、清华集训、THU/PKUWC/SC、Codeforces

- 及时停下手中的娱乐活动
- 发现状态不对，不需要急于开始训练，可以调整
- 欢迎找我聊天，不过和自己聊天是最好的方式
- 避免一些胡思乱想

知识点讲解

- 多数不会的重要知识点，或者重要且难度上限高的知识点（个别同学不会的重要知识点不讲，但同学要在课下积极学习！）
- 讲解应用
- 分析知识点在比赛中的重要程度

杂题选讲

- 体量小而注重思维模式的题目
- 培养大家打破思维僵局，开拓思考方式的“头脑风暴”能力

模拟比赛训练

- 自主抽出时间完全仿真比赛（是否可以?）
- 在课上对比赛进行回顾和讲解
- 不局限于讲正解，还有应该采取什么比赛策略？部分分的作用？如何从部分分得到启发？

杂谈

欢迎提问！

大家已经做过 NOI2021 的题目，现在我们就跳过做题的环节直接解析比赛。我将先讲述每一道题从部分分开始一直做到满分解法的思路，并演示数据范围是如何启发我们的。

最后，我将综合讲述如果这一场比赛再次降临，我们应该采取何种策略应对比赛。

题目

- 轻重边
- 路径交点
- 庆典

给大家一些时间回顾比赛的题目。

题意简述：给定一棵 n 个节点的树，初始所有边为轻边。两种操作：给定一条树链，将与树链上点相连的边改为轻边，再将树链上的边改为重边；给定一条树链，查询有多少重边。

题意简述：给定一棵 n 个节点的树，初始所有边为轻边。两种操作：给定一条树链，将与树链上点相连的边改为轻边，再将树链上的边改为重边；给定一条树链，查询有多少重边。

初步分析：首先，这是一道数据结构题。注意到这是 NOI 的 D1T1，而 NOI 是非常严谨的比赛，T1 一般难度不大，所以这道题的正解应该不需要用到很复杂的数据结构。因此，不必想得过于复杂，当思路变得越来越复杂时，应该及时改正方向。

题外话

当我们做数据结构题时，常常陷入一个怪圈。我们会先想出一个不严格的算法，接着发现这个算法有的地方不符合题目要求，于是我们打上一些补丁，接下来又会不断重复这个打补丁的过程，甚至是发生在花了很长时间写完代码之后。最终，我们往往会调试失败或者根本没有想出正确解法。发生这个现象的根本原因，在于我们对题目性质了解得还不够透彻，在迷迷糊糊的状态下匆忙开始写代码，最后浪费了大量时间；同时，在算法随着补丁变得越来越复杂后我们已经基本丧失了对算法的掌控能力，开始不知道每一步在做什么。这时应该采取的正确方法是：果断放弃越来越复杂的算法，重新思考一遍题目的本质，从头推出一个简洁的算法；同时，不要轻易尝试去写一个自己还没完全明白在做什么的算法。

不难首先想到一个显然的 $O(n^2)$ 暴力做法。观察数据范围，这可以得到 30 分。

又在数据范围中找到一个性质 A：树是一条链，很快得到一个区间赋值的做法：如果 $[l,r]$ 变成重边，就将 $[l,r-1]$ 区间赋值为 1，$l-1,r$ 赋值为 0，线段树即可。得到 50 分。

极限数据范围是 10^5 级别，那么应该是 poly log 的算法或者根号算法。鉴于低难度数据结构题对 poly log 做法比较偏爱（题外话：从 OI 比赛的命题规律来看，根号算法往往应用在复杂的数据结构问题中，因此用到根号算法的题目难度偏高），可以先往前者方向想。

此时我们有几个方向可以走：继续思考性质 B；尝试根据性质 A 的做法得到启发想出正解；另外还有注意到这题的操作与 LCT 的 access 类似，有可能可以对 LCT 加以改进做出这道题。

需要特别说明的是，我个人不建议大家采取第三种方向，即改进 LCT。因为 LCT 本身已经是一个系统化的、复杂的数据结构，很少能够被成功改造。在这道题中我们很容易发现单纯的 LCT 是不足以解决问题的，比如 access 不支持换根的操作，而要用 makeroot 就会打乱轻重边。

当然，实际上这个选择也是可行的。不过最终的做法更加复杂难以调试，不适合作为 T1 的正解。因此这里略去。

我们先来看看思考性质 B 会如何，性质 B 是只查询单个边。如何快速判定一条边是不是重边？来看看重边何时会出现。不难发现，当两个相邻的点同时出现在第一个操作中时，它们之间的边就变成重边；而如果有一次操作只包含其中一个点而不包含另一个点，那么这条边就成为轻边。那么做法就比较明显了：我们记录当前每个点 x 最迟被包含在第 a_x 个操作中，查询一条边 (x, y) 时，如果 $a_x = a_y$，这就是重边，否则是轻边。

而修改也是非常简单的，每次需要对一条树链区间赋值：树剖 + 线段树即可。复杂度 $O(n \log^2 n)$，得到 70 分。

事实上，当我们想到这个做法之后，很容易把这个做法改造成正解：我们只需要考虑对一条树链统计有多少相邻的 $a_x = a_y$ 的点对即可。这可以这么在线段树上维护：对于区间 $[l, r]$，维护三个值：区间内相邻数字相同的对数，a_l，a_r，更新和区间覆盖都很容易。于是我们就得到了一个简单的 $O(n \log^2 n)$ 做法，通过了此题。

再看看第二个方向：尝试根据性质 A 的做法得到启发想出正解。性质 A 的做法用了线段树，想要把线段树搬到树上，自然要先考虑树链剖分了。

首先熟练地将对一条链的操作转化为 $O(\log n)$ 条重链上的操作，树链端点之间的做法比较容易。对于一条重链，我们对其中的点类比性质 A 的做法，设置一个 0/1 变量 t_0，为 1 就代表这个点和重儿子之间连了重边，否则置 0，并额外用至多两个变量 t_1, t_2 表示和哪些轻儿子之间连了重边（这一步需要经过一定思考，首先发现我们需要知道它和哪些轻儿子有重边，其次发现最多只有两个这样的轻儿子）。现在，我们想要将重链中的一个区间全部改造为重边，那么就对对应区间的 t_0 赋值为 1，t_1, t_2 赋值为 0（表示这样的轻儿子不存在）。

有了 t_0 之后，区间查询也是容易的。于是我们就得到了正解，复杂度 $O(n\log^2 n)$。

最终我们通过不同的思考方式得到了两种正解。值得一提的是，偏简单题的做法往往比较多样，因此这题还存在许多其它做法。

同时回顾一下我们的做题过程，可以发现通过性质 B 得到的做法转化为正解更容易，所以在做题时，如果想到了一个部分分做法，可以不必着急将其转化为正解，而是完整浏览部分分，找到最合适转化为正解的做法。当然，这也不是绝对的。

题意简述：给定一张 n 层有向图（第一层与最后一层点数均为 n_1），称 n_1 条点不相交的路径为一种方案，k 为该方案的权值，其中 k 为路径交叉的次数。求权值为偶数的方案比权值为奇数的方案多多少。

初步分析：首先，一个熟练的 OI 选手应该对“权值为偶数-权值为奇数”非常敏感，为什么要求这个数呢？隐约可以猜到答案，因为标算想要统计的其实是方案的带权和，而权值正是 $(-1)^k$。而这样涉及交点、又只要考虑奇偶性的题目，在 OI 里对应的知识点几乎是唯一的：逆序对、行列式（类似地，题目特地只让你考虑“模 2 意义下的方案数”，那基本是这个知识点了）。

题意简述：给定一张 n 层有向图（第一层与最后一层点数均为 n_1），称 n_1 条点不相交的路径为一种方案，k 为该方案的权值，其中 k 为路径交叉的次数。求权值为偶数的方案比权值为奇数的方案多多少。

初步分析：首先，一个熟练的 OI 选手应该对“权值为偶数-权值为奇数”非常敏感，为什么要求这个数呢？隐约可以猜到答案，因为标算想要统计的其实是方案的带权和，而权值正是 $(-1)^k$。而这样涉及交点、又只要考虑奇偶性的题目，在 OI 里对应的知识点几乎是唯一的：逆序对、行列式（类似地，题目特地只让你考虑“模 2 意义下的方案数”，那基本是这个知识点了）。

一些同学可能认为这道题是 LGV 引理的模板题，没什么值得讨论的。但我觉得可以不这么理解，即使完全抛开 LGV 引理，有水平的选手也可以在考场上独立推导出它。如果认为这是 LGV 引理的模板题，那是不是意味着我们要学习很多生僻的算法来应对可能的类似题目呢？但应该说，这样的出题风格目前并不是 OI 的主流。

乍看之下，不存在什么容易得到的暴力做法，先看看部分分。$k=2$ 的这一档看上去比较宽松，不妨先进行尝试。

稍加思考，发现对于一个两层的图，每一种路径方案都可以看成是一种排列，而交点的个数也就是排列的奇偶性。那么所求的答案，其实就是找到所有可以选择的排列，再将它们的权值相加。

乍看之下，不存在什么容易得到的暴力做法，先看看部分分。$k=2$ 的这一档看上去比较宽松，不妨先进行尝试。

稍加思考，发现对于一个两层的图，每一种路径方案都可以看成是一种排列，而交点的个数也就是排列的奇偶性。那么所求的答案，其实就是找到所有可以选择的排列，再将它们的权值相加。

上述的思考让我们容易联想到行列式的完全展开式：

$$\det A = \sum_{p} (-1)^{\operatorname{sgn}(p)} \prod_{i=1}^{n} A_{i,p_i}$$

可以发现，给定邻接矩阵 A，某一个排列 p 对应的路径方案是否可行正好对应于 $\prod_{i=1}^{n} A_{i,p_i}$ 是否为 1。而行列式的系数又自然地为我们解决了带权的问题，于是得到结论：$\det A$ 即为所求答案，复杂度 $O(n^3)$，得分 35 分。

接下来，尝试一下性质 A、B 同时存在。条件是所有 n_i 相等，且方案最多只有一个，只要找到这个方案是什么，再计算逆序对数就得到答案了。考虑下这个方案会长什么样：那一定是每两层之间都找出了 n 条边，使得它们两两没有公共点。这似乎有些眼熟，进一步想想，这不就意味着找到一个完美匹配吗？于是算法就产生了：每两层之间都计算出最大匹配，就可以得到方案了。得分 55 分。（题外话：如果条件 A 不满足但条件 B 满足，还可以在整张图上用最大流找到方案，不过没有这档部分分，可能是出题人没想到这个做法）

不过，“只存在一个方案”这个条件太苛刻了，最大匹配也和计数题画风相差太大，不应该从这里得到启发找出正解。再来看看性质 A，每层的 n 相同会如何？最大匹配的做法虽然难以拓展，但思路可以借鉴：既然每一层能单独做匹配，是不是意味着 n 相等时不同层之间有一定的独立性？

不难发现确实如此：只要给每一层选出了一个完美匹配的方案，这些方案总能组合成路径方案，并且，路径方案的权值恰好是这些完美匹配对应排列的权值的乘积。有了良好的独立性做法就显然了：每一层的做法之前已经讨论过，求行列式即可。不同层组合起来，则由于独立性只需要把答案相乘即可。求 n 个行列式就做完了这一题。

题外话：什么是独立性？

独立性其实对应着组合数学题目里一种显著降低求解复杂性的技巧，很多难题都会用到这一点。具体解释为：现在想求所有合法方案 S 的权值和 $\sum_{S\ is\ legal} w$，而每个合法方案 S 都可以被分解为 n 个部分 $S_1, S_2, \ldots, S_n$，并满足下面两个性质：

1. S 合法当且仅当 $S_1, S_2, \ldots, S_n$ 都合法。
2. 令 S_i 的权值为 w_i，有 $w = \prod w_i$。

那么，我们可以将原问题分解为 n 个部分，分别求解合法的 $S_1, S_2, \ldots, S_n$ 的权值和：

$$\sum_{S\ is\ legal} w = \sum_{S_1, S_2, \ldots, S_n\ is\ legal} \prod w_i = \prod_{i=1}^{n} \left(\sum_{S_i\ is\ legal} w_i \right)$$

分别求出 n 个部分的答案再相乘即可。得分 75 分。

回到题目，能否从这个做法得到启发想到正解呢？

由于行列式乘积不变的性质，我们不难想到将上一个做法的常数变得更小：直接求邻接矩阵的乘积再整体求行列式即可。我们不禁发现由于题目特别保证了头尾点数相同，这个做法对 n 不相等的情况仍然适用。我们自然希望知道直接照搬做法能否管用。

回到题目，能否从这个做法得到启发想到正解呢？

由于行列式乘积不变的性质，我们不难想到将上一个做法的常数变得更小：直接求邻接矩阵的乘积再整体求行列式即可。我们不禁发现由于题目特别保证了头尾点数相同，这个做法对 n 不相等的情况仍然适用。我们自然希望知道直接照搬做法能否管用。

由于测试这个方法的代价比较小，只要稍微改动一下代码测一测大样例即可，我们可以不加证明直接这样做。最后我们将发现这个做法能够通过大样例，如果采纳了这个做法，就可以得到满分。（因此，当一个结论看上去希望很大而且测试起来非常方便，就不妨直接写代码进行验证）

那么，有没有严格一些的方法呢？

事实上，如果我们在一开始思考一下 $n \leq 10$ 的部分分，或许能更快想到标准解法。这个数据范围似乎在引导我们向阶乘的复杂度思考，而因为中间点的数目仍然是 100 的级别，只能在头和尾考虑 10 的作用。枚举每条路径的起点和终点如何呢？画图进行考虑，假设有两条路径，起点分别为 $i, j(i < j)$，终点为 p_i, p_j，那么如果 $p_i < p_j$，两条路径就必须相交偶数次！对 $p_i > p_j$ 的情况类似。我们得到一个重要结论：两条路径相交次数的奇偶性只和路径起点与终点的相对位置有关！

那么算法就来了：枚举所有可能的起点和终点排列，假设是 (i, p_i)，那么奇偶性就已经确定，只需要统计这一排列的方案数。由矩阵乘法的定义，我们自然可以得到 $\left(\prod A_k\right)_{i,p_i}$ 是 i 到 p_i 的路径数。那么我们能否直接将它们相乘呢？

可以发现，如果直接计算 $\prod_{i=1}^{n}\left(\prod A_k\right)_{i,p_i}$ ，那么 i 到 p_i 的路径可能和 j 到 p_j 的路径可能相交。此时，一个较常见的套路是把它们的贡献消去：对于一个有路径相交的方案，找到最上层且标号最小的交点，将两条路径之后的轨迹交换就可以得到一个新方案。可以发现这是将所有有路径相交的方案一一配对，并且配对的两个方案由于交换了路径，贡献和为 0。因此得到结论：路径相交的方案贡献和为 0，不影响计算。（这是一个常见的证明策略，想要证明一群方案贡献和为 0，可以找一种方法将它们两两配对，且每一对贡献和都为 0）

于是我们的式子就是：

$$\sum_{p}(-1)^{\mathrm{sgn}(p)}\prod_{i=1}^{n}\left(\prod A_k\right)_{i,p_i}$$

这实在太明显了。直接改用行列式就这个值，就得到了正解。

题意：给定一张满足一个性质的有向图，每次询问给定起点和终点，以及不超过 2 条临时加入边，求起点到终点可能经过的点的数目。

题意：给定一张满足一个性质的有向图，每次询问给定起点和终点，以及不超过 2 条临时加入边，求起点到终点可能经过的点的数目。

初步分析：看到关于可达性的描述，首先自然地将强连通分量缩为一个点，那么原图就成为了一张有向无环图。不考虑特殊性质的情况下，想要在有向无环图上回答一系列带修改的问题往往是非常困难的。在这道题中，不利用性质的暴力做法复杂度为三次方级别，只能得到 16 分。因此，做这一题最重要的是考虑如何有效地利用题目所给的性质。

回顾一下性质：如有 $x \to z, y \to z$，那么 $x \to y$ 或 $y \to x$。可以这么描述这个性质：如果两个点有着共同的后继，那么它们之间有着严格的序关系；从另一个角度看，就是说一个点的所有前驱都有序关系。这样，如果我们进行拓扑排序，所有的前驱就会形成一条链。

对这一情形稍加思索不难发现，拓扑排序后所有点将会形成一个树状的结构。对于每一个点，它所能到达的点就是它子树内的点。在 DAG 转化为树状结构之后，问题就变得可做很多了。

现在我们可以直接得到一个平方做法和 $k=0$ 的做法，得分 44 分。

$k=1$ 的情形看上去不复杂，可以尝试是否能分类讨论。假设起点、终点为 x,y，新添加的边为 $a\to b$。首先是不经过新添加的边的简单情形，以及经过一次边的情形。如果经过一次边，那么路径形如 $x\to\cdots\to a\to b\to\cdots\to y$，也就是说，树上的链 $(x,a),(b,y)$（如果存在）的点也是可达的。此时，我们还要特别关注容易被遗漏的情况——多次经过边 $a\to b$。此时，如果链 (b,a) 存在，那么这条链中的点也能到达。于是，通过分类讨论我们解决了 $k=1$ 的情形。得分 64 分。

$k = 2$ 的情形当然可以继续尝试分类讨论。不过我们很快会发现情况比较复杂，难以讨论清楚。一种方法是理清思路，清晰地继续分类讨论。另一种则是把这个问题更加一般化，尝试引入对较大的 k 也适用的新算法来解决。

$k=2$ 的情形当然可以继续尝试分类讨论。不过我们很快会发现情况比较复杂，难以讨论清楚。一种方法是理清思路，清晰地继续分类讨论。另一种则是把这个问题更加一般化，尝试引入对较大的 k 也适用的新算法来解决。

我们先试试后一种选择。现在，假设新加入的边是 $a_1 \to b_1, a_2 \to b_2$。那么不难发现除 x, y, a_1, b_1, a_2, b_2 六个点外，其余的可达点则都处在这六个点中某两点之间的树链中。

$k=2$ 的情形当然可以继续尝试分类讨论。不过我们很快会发现情况比较复杂，难以讨论清楚。一种方法是理清思路，清晰地继续分类讨论。另一种则是把这个问题更加一般化，尝试引入对较大的 k 也适用的新算法来解决。

我们先试试后一种选择。现在，假设新加入的边是 $a_1 \to b_1, a_2 \to b_2$。那么不难发现除 x, y, a_1, b_1, a_2, b_2 六个点外，其余的可达点则都处在这六个点中某两点之间的树链中。

既然问题涉及孤立点之间的树链，那么我们就建立起关于六个点的虚树，加上这两条边，我们在这棵虚树上用暴力找出所有虚树上的可达点。最后对于每一对有树链相连的可达点，将树链上的点都算作可达点即可。

虚树的做法显然可以用到较大的 k 上，对于 $k=2$ 的情形来说，有些大材小用，同时写虚树也不容易。是否可以继续分类讨论呢？

首先考虑可能的路径的一般形式，那么无非是若干段树边、一段 $a_1 \to b_1$ 或 $a_2 \to b_2$、再来若干段树边...... 如此往复。

虚树的做法显然可以用到较大的 k 上，对于 $k=2$ 的情形来说，有些大材小用，同时写虚树也不容易。是否可以继续分类讨论呢？

首先考虑可能的路径的一般形式，那么无非是若干段树边、一段 $a_1 \to b_1$ 或 $a_2 \to b_2$、再来若干段树边...... 如此往复。

可以发现走树边的情形无非几类：

$(x,a_1),(x,a_2),(x,y),(b_1,a_1),(b_1,a_2),(b_1,y),(b_2,a_1),(b_2,a_2),(b_2,y)$。注意到 x,y 只会出现在起点和终点，我们可以先枚举 x 的下一个点和 y 的上一个点。在这之后，继续枚举是否有 $a_1 \to b_1$ 之后再走 $a_2 \to b_2$ 的情况，是否有 $a_2 \to b_2$ 之后再走 $a_1 \to b_1$ 的情况. 每一种情况确定后，操作就变成如果所需要的树链都存在，那么这些树链上的点均为可达点。实际处理难度不算难，但是这样的分类讨论题在开始讨论之前一定要把情况分类清楚，不能让自己的思维产生含糊。否则，几乎是分类讨论不出来的。

对于自己分类讨论能力不自信的同学，也不妨采取把问题一般化的方法，找到一个对更大的 k 也可行的算法，这就主要看个人偏向了。

首先需要说明的是，NOI 比赛和平时比赛相比可能更加特殊，特殊在于难度几乎一定是按照顺序排列的。因此，尽管平时我们需要善于观察题目选择开题顺序，但是 NOI 比赛可以直接采取按顺序进行的办法。

同时，根据经验 NOI 的第一题几乎一定是可以比较容易解决的中档题，应此做题就要注意两点：尽可能快地找到完美做法，同时不要对这题产生畏难心理，一旦觉得自己的思路越来越复杂了，果断换一个思路。

在解决完第一题后，我们再将第二题和第三题分别看一遍。第二题只有一层的情形是一个比较显然的暴力，并且和最终范围比较相似，可以先把这个部分分写完作为打底。而对于第三题来说，直接暴力的做法分数很低，而且明显地没有扩展性，因此可以适当选择放弃，去攻克第二题的其它部分分寻找启发或是思考第三题的性质能带来什么结论。

第二题比较关键的地方在于 n 相同的做法以及 $n \leq 10$ 的部分。如果能够得到这两个部分分中的一个，那么得到正解就不太难。而性质 B 比较显然是一个对正解没什么启发的部分，可以选择跳过。当然，如果最终没能思考出正解，那么写出这一档部分分也是“贪得无厌”的一种选择。而第三题的关键在于树形结构的发现，此后的策略就正如对第三题的讨论所述，可以根据自己的情况有多种选择。

希望本节课讨论的内容对同学们有所启发！