

IOI2013 解题报告与参赛总结

南京外国语学校 许昊然

Contents

1 Dreaming 解题报告	2
1.1 题目大意	2
1.2 算法讨论	2
2 Artclass 解题报告	3
2.1 题目大意	3
2.2 算法讨论	3
3 Wombats 解题报告	3
3.1 题目大意	3
3.2 算法讨论	4
4 Cave 解题报告	4
4.1 题目大意	4
4.2 算法讨论	4
5 Robots 解题报告	5
5.1 题目大意	5
5.2 算法讨论	5
6 Game 解题报告	6
6.1 题目大意	6
6.2 算法讨论	6
7 参赛总结	6

1 Dreaming 解题报告

1.1 题目大意

给定一个包含 n 个结点的森林($n \leq 10^5$), 边上有权值。给定一个值 L , 你需要增加若干边权均为 L 的边, 使得:

- 得到的新图是一棵 n 个结点的树
- 这棵树的直径尽可能小

问最小可能直径是多少。

1.2 算法讨论

经过一些观察, 我们可以发现以下性质:

1. 如果某个结点是我们新加的某条边的端点, 那么我们称这个点为“接点”。

可以证明存在一个最优解使得在每个原图中的树里, 都有且仅有一个“接点”。

证明: 如果有一个最优解使得在某个树中有超过一个“接点”, 我们不妨取其中任意两个, 设为 A 与 B 。我们不妨定义 $dis(T)$ 表示结点 T 所在树中, 距离结点 T 最远的点的距离。不妨设 $dis(A) \leq dis(B)$, 那么我们将接到结点 B 的边全部改为接到结点 A 上, 可以发现, 直径不会增加。如此往复, 直至这棵树中只剩一个接点, 此时解没有变差。

2. 可以证明存在一个最优解, 使得对于原图中每个树, 这个树的接点 T 必定是 $dis()$ 值最小的点。

证明: 如果有一个满足上个性质的最优解中, 某个接点 T 的值不是所在树中 $dis()$ 最小的结点, 不妨设 $dis()$ 最小的结点为 T' , 我们把所有接到结点 T 的边都改为接到节点 T' 上, 可以发现, 直径不会增加。

3. 有了前两条性质, 我们现在需要考虑的就仅仅是这些“接点”的连接方式了。容易证明, 最优的连接方式一定是: 选择一个接点, 其他所有接点都向这个接点直接连边。

这个证明比较难写……不过感性理解的话其正确性是显然的, 就略过了……

有了上文的分析, 算法已经非常显然了。首先我们需要求出原图中每个树的直径, 并求出原图的每个树中 $dis()$ 的值最小的结点。

这一步只需对每个结点用树形dp求出其向下最长延伸、次长延伸的长度, 以及向上最长延伸的长度即可 $O(n)$ 完成。方法十分经典, 在此不再赘述。

我们不妨用 $D[i]$ 来表示我们求出的树的 $dis()$ 的最小值。接下来, 我们考虑枚举那个“其他接点都向它连边”的接点, 那么当前方案的答案可以利用 D 数组的最大的三个值 $O(1)$ 计算出来(具体细节留给读者)。

最后还有一个小细节, 上文所算的答案仅仅考虑了包含我们新加的边的路径, 因此最后还应该和原图中各个树的直径取 max , 才能得到最终答案。

最终复杂度 $O(n)$ 。

2 Artclass 解题报告

2.1 题目大意

给定一张不超过500像素x500像素的图片，要求识别这张图片是4种油画作品风格中的哪一种。具体每种风格的图片示例请参见原题面。

题目中的每种风格都向选手提供了9张作品示例（总计36张）。要求选手在官方测试数据中达到尽可能高的正确率，达到90%正确率即可得到满分。

2.2 算法讨论

这是一道图像识别类型的开放题。这类图像识别题的一般思路是，对给定图片找一些区分度较强的特征值来进行区分。

一种比较好的策略是使用类似机器学习的方法：如果找了 n 个特征值，就得到了一个 n 维空间中的点，我们先预存若干样本得到的点，询问时选择和样本点距离最接近的那个样本的油画类型作为答案。但我考场上没有选择这个方法，主要原因是觉得36幅图片实在太少了，如果18张作为训练集18张作为测试集的话，平均每种风格在训练集中只能出现4次，测试结果必定非常不准确；而且这个方法也不是很好写。

我采用的方法是选择一些特征值后，手工写判定流程来进行判定。

经过观察题目提供的图片，容易发现，这四种油画风格区别主要在于色块的大小和色块内色彩的差异程度。

因此，对于图片中的每个像素，我考虑以其为中心的 $7*7$ 矩形中的其他像素。考虑其他像素与其的颜色差异（计算红绿蓝这三种颜色的差的平方和），计算这些值大于1000的比例、大于2000的比例、大于3000的比例、大于4000的比例、大于5000的比例以及标准差的平均值作为图片的特征值。

我写了个脚本，对于给定的36张图片计算这些特征值，然后找规律，利用这6个特征值进行判定。这个算法实际效果是非常好的，我第一次提交就得到了满分，而且正确率达到了98%，远高出题目要求的正确率。

3 Wombats 解题报告

3.1 题目大意

有一个 $R * C$ 的网格，其中所有的竖向边只能从上向下单向通行，而横向边可以双向通行。（ $R \leq 5000, C \leq 200$ ）。每条边都有一个初始权值。

要求支持：

- 修改一条边的权值
- 查询从网格最上面一行的某一点到最下面一行的某一点的最短路的长度

修改操作不超过500次，查询操作不超过 $2 * 10^5$ 次，时间限制20秒。

3.2 算法讨论

因为竖向边只能从上向下单向通行，这个题目具有很明显的阶段性。朴素的dp可以做到 $O(RC)$ 每次询问。

进一步观察可以发现，因为竖向边都是单向的，这里的最短路是满足区间合并性质的。如果我们有两个 $X * C$ 的网格中，最上面的行和最下面的行两两之间最短路的长度，那么我们显然可以在 $O(C^3)$ 内合并它们，得到这两个网格拼起来后的情况。

我们不妨用线段树维护 R 这一维，那么复杂度就是 $O(C^3 * \log R)$ 每次修改， $O(1)$ 每次查询（因为查询只查询整个网格的情况）。

但这么做对于题目的规模依然是不够的。我考场上只想到了这一步，因此只得了55分。其实因为网格是平面图，因此合并时是满足单调性的。

我们固定上半部分网格的起点，然后考虑拼接处每个点能作为哪些端点的最优答案。可以证明，以其为最优答案的终点必定是一个区间，也就是转移是有单调性的。因此，只要套用经典的1D/1D动态规划优化方法，用一个单调栈+二分来维护最优决策即可把合并复杂度优化到 $O(C^2 * \log C)$ 。

最终复杂度是，预处理 $O(C^2 * R * \log C)$ ，修改 $O(C^2 * \log C * \log R)$ ，查询 $O(1)$ ，足以通过本题数据。

4 Cave 解题报告

4.1 题目大意

在一个洞穴里有 n 个开关和 n 扇门（ $n \leq 5000$ ）。每个开关都对应着一扇门，但你不知道哪个开关对应哪扇门。每个开关都有一个正确状态，当开关处于正确状态（关或开）时，对应的门就会打开，否则就会关闭。当然你也不知道每个开关的正确状态。

现在你可以每次可以设置 n 个开关的状态，然后走进洞穴查看第一扇关闭的门是第几道门。你的任务是用不超过70000次询问，确定所有门与开关的对应关系，以及每个开关的正确状态。

4.2 算法讨论

首先朴素做法是比较简单的：我们考虑先随意设置开关的状态，看一下第一扇关闭的门。然后我们每次翻转一个开关的状态，再去看一下。如果此时第一扇关闭的门的序号变小了或增加了，那么这扇门必定是因为我们的翻转而关闭或开启的，我们从而得到了一个门的对应开关以及它的正确状态。

我们从第一个开关到最后一个开关都这么试验一遍，就必定能确定至少一扇门（因为最初的第一扇关闭的门也肯定对应了其中一个开关），如此做至多 n 次，即可确定所有门的状态。询问次数是 $O(n^2)$ 的。

当然上述算法对于题目的要求是远远不够的。我们不妨思考一下上述做法的核心，其核心思想是，如果我们翻转了当前状态的某些开关，并导致了第一扇关闭的门的序号变化，那么我们就能够确定一扇门的对应开关在我们翻转的那些开关中。

因此，我们考虑不要一次只翻转一个开关，改成一次翻转一半的开关。这样，我们就能够确定一扇门的控制开关是否位于我们翻转的开关中，而不管是或不是，我们都能将这扇门可能的控制开关数量缩小一半，如此往复，只要 $O(\log n)$ 次询问就必定能确定一扇门的对应开关及其正确状态。

这样一来，询问次数被降低到了 $O(n \log n)$ ，足以满足题目要求。

5 Robots 解题报告

5.1 题目大意

地上有 T 个玩具，每个玩具有一个重量 W_i 和一个体积 S_i ($T \leq 5 * 10^5$)。有一些机器人，每个机器人每分钟可以捡起一个玩具并把它放进柜子。

有两种机器人：

- 弱机器人，每个这种机器人自身有一个参数限制 L ，代表其能捡起重量小于 L 的任意玩具（不管体积是多少，各个机器人的 L 值不一定相同）
- 小机器人，每个这种机器人自身有一个参数限制 L ，代表其能捡起体积小于 L 的任意玩具（不管重量是多少，各个机器人的 L 值不一定相同）

现在有 n 个小机器人和 m 个弱机器人。问你在最优方案下，要多少分钟才能收拾好所有的玩具。（ $n + m \leq 5 * 10^4$ ）

5.2 算法讨论

看如此巨大的规模，应该会存在一个贪心策略来设计最优方案，否则这题就没法做了。但我当时在考场上想了很久都没有想法，最后终于想到了一个自己没找到反例的贪心策略，就写了一下，结果AC了。但至今我都没有证明我的贪心策略的正确性。官方做法是一个正确性更加直白的贪心策略。下文中这两个做法我都会说一下。

我的贪心策略是这样的：我们先考虑所有弱机器人。我们把每个玩具 (W_i, S_i) 表示为坐标 (W_i, S_i) ，把所有的弱机器人的限制 L_i 表示为直线 $x = L_i$ 。

这时我们发现，直线把平面分成了“一条一条”的区域。我的策略是对于每个弱机器人，只要它前面的“一条区域”中有玩具，就应该取走纵坐标最高的那个玩具；如果没有任何玩具，就考虑更靠前的一条区域，如此往复。

对小机器人的策略是相同的，只不过横坐标和纵坐标换了一下。

用STL可以把这个算法优化到 $O(n \log n)$ 。

这个策略我没有严格证明正确性，但也暂时没找到反例（事实上它确实AC了）。下面是正确性更加直白的官方做法。

官方做法是这样的：首先这个问题显然是满足二分性质的，我们二分答案，不妨设答案

为 Ans ，那么我们把每个机器人都复制 Ans 遍。这样问题被转化成了一个覆盖问题，每个机器人能覆盖一个玩具，问能否覆盖所有的玩具。

我们依然把每个玩具 (W_i, S_i) 表示为坐标 (W_i, S_i) 。我们不妨按照弱机器人的 L 值从大到小的顺序考虑。我们容易发现，因为横坐标在不断减小，因此限制值比当前横坐标更大的“弱机器人”其实互相之间已经没有区别了，它们都能覆盖没有被覆盖的玩具中的任意一个。

而“小机器人”相比“弱机器人”，这时就“无用”很多了，因为它们只能覆盖纵坐标不超过某个值的玩具。

因此，我们按横坐标从大到小加入事件，如果当前事件是一个弱机器人，就把备用的弱机器人数目加一；如果当前事件是一个玩具，那么我们应该尽可能用“小机器人”去覆盖，而且显然应该用“最差”（参数最小）的能覆盖它的小机器人去覆盖。如果小机器人覆盖不了当前玩具，那么只能用一个备用的弱机器人来覆盖了，备用弱机器人数目减一。如果发现某个时刻备用弱机器人不够用了，那么就无解。

上述算法的复杂度是 $O(n \log^2 n)$ ，但正确性很直观。

6 Game 解题报告

6.1 题目大意

有一个 $10^9 * 10^9$ 的矩阵，每个方格中有一个数字。初始所有数字都是0。

要求支持以下两个操作：

- 修改一个方格中的数
- 给定一个矩阵，查询这个矩阵中所有数的最大公约数

修改数目不超过22000个，查询数目不超过 $2.5 * 10^5$ 个，时间限制13秒，内存限制230MB。

6.2 算法讨论

这道题是非常经典的题目了，相信任何有省队水平的同学都会做。可以使用trie树套trie树做到 $O(\log^2 n)$ 每次操作，但这样内存也是 $O(n \log^2 n)$ 的，会爆内存。我们可以把里层的trie树改成splay树，这样内存就降低到 $O(n \log n)$ 了，可以接受。

这道题还有一些很好玩的做法，比如使用陈立杰同学在CTSC时候介绍的重量平衡树，修改操作的复杂度可以优化到期望 $O(\log n)$ ，但查询的复杂度依然是 $O(n \log^2 n)$ ，对这题修改较少，查询很多的情况来说意义不大。

还有一些不太靠谱的做法，比如用k-d树之类的，对于随机数据有效，但对精心构造的数据就要退化了。

7 参赛总结

能作为中国队的一员参加IOI2013，我感到十分荣幸，这次参加IOI的经历将是我难忘的回忆。

这次IOI在澳大利亚布里斯班举行。行程安排与往年大致相同：7月6日到达昆士兰大学，7日进行试机赛和开幕式，8日是第一试，然后9号去黄金海岸玩，10号进行第二试，11号去澳洲动物园玩，最后7月12日闭幕式。

布里斯班给我的第一感觉是环境优美，绿化做的非常好，随处可见草坪、树林，因此空气也很好，很适合人类居住。我在布里斯班期间从未有过类似“空气灰蒙蒙的”这种在北京或南京很常见的感觉，而且每天晚上都能看到星星（说实话我在南京城区已经很少能看到星星了，在北京的几天更是从未看到过），可见其环境之好。听领队孙辉老师说中国70年代时候也是蓝天白云，后来搞工业逐渐污染了。然后说污染对人到底有多大影响还不知道，因为中国吸着污染空气的这一代人还没老呢。但我觉得历史上血淋淋的事实已经足够了，比如伦敦雾霾，比如日本的汞中毒事件。中国有城市也已经开始出现雾霾了，最近又曝出了镉大米事件，英国和日本的前车之鉴值得我们警惕。最近好像在宣扬科学发展观以人为本，那就更应该好好保护环境，保障人民生命安全，真正做到以人为本，而不是靠找所谓“专家”发表“镉大米吃一两年死不了”这种无耻言论来糊弄民众。毕竟GDP是不能吃的，每天吃着镉大米的话，GDP再多又有什么用。这次澳大利亚之行，让我深深感到了中澳环境保护方面的差距，中国距离真正的富强还有很长一段路要走。

然后说说比赛吧，这次比赛规则与去年基本相同，不同点是取消了token制，改为每道题最多100次提交，并且每个提交都会在完整测试数据上跑并返回结果。也就基本等于完全取消了评测次数限制，毕竟只要你不干用提交来获取数据之类的事情，100次提交是肯定用不完的。但是这也为第一试的混乱情况埋下了祸根。

第一试我发挥的并不十分理想，有评测系统挂掉的原因（第三题wombats每组数据时限20秒，且有100组数据，而设定的提交间隔仅为1分钟，导致评测系统卡死），但更多还是我的实力和心态问题。第一试第一题dreaming是一道传统题，这道题并不难，在快要一小时时候我解决了这道题。第一试第二题artclass是一道开放题，要求识别一张油画是4种风格的哪一种。我找了几个特征值写了判定，抱着试试看的心态提交了一次，竟直接AC了，而且正确率达到了98%（题目只要90%就给满分），这应该是本次IOI中最令我自豪的一道题目了。第三题wombats是一道数据结构题，其中关键部分要用到一个单调性，这时时间大约是一个半小时，距离比赛结束还有将近4小时（比赛因为评测系统挂掉而延迟了半小时，虽然评测系统一直到比赛结束都没有修好），我本来是完全有机会做出这道题的。但当时因为被评测系统搞的很火大，加上主持人不停的发公告，心里很烦，而没有完全集中精力思考；也因为类似题目做的太少，到最后都没能自己推出这个单调性。现在想想，如果当时冷静的接受现实，直接这道题当作传统OI题来做，不再折腾那个已经完全卡死的评测系统，很可能就是另外一种结果了。没有做出这道题目是我这次参赛最大的遗憾，给我的教训是任何时候都一定要保持良好的心态，不能被不可抗的外部环境影响自己的心态。

第二试题目相对比较简单，但我做的十分惊险。第一题Cave是比较简单的交互题，但因为我的思维缺陷，花了很久才做出这道题目。然后我看了第三题Game，觉得是不难的数据结构题，只需trie树套splay即可。于是写了第三题，但因为一个小bug一直爆零，到距离比赛结束只有1个小时时候才发现了bug并AC了这道题。看完第二题Robots后感觉是某种贪心，但想了一会并没有什么特别靠谱的想法，有一个贪心策略暂时没找到反例，由于此时留给我的

时间已经非常紧张了，我直接写了那个策略的朴素算法。提交上去后，得到了76分，而且大数据都是超时而不是错误，这让我相信这个贪心策略是正确的。我用数据结构优化了那个程序，在离比赛结束还有15分钟时，完成了优化后的程序，但提交依然只得到了76分。于是我开始查错，在离比赛结束还有大约3分钟时候，发现了一个边界情况的细节错误，改正后提交，AC了，此时离比赛结束只有不到30秒时间了。第二试的比赛我做的十分惊险，尤其是在做Robots这题时候，当时局势非常不利，此时如果产生了放弃念头，那么结果必然是惨挂。这也从另一面说明了保持良好心态的重要性。

赛后开总结会时提到过NOI是否也要采用IOI赛制的问题，我觉得这弊大于利。IOI赛制相对NOI赛制的好处还是很多的，比如说，现行OI比赛制度中选手只有一次机会，写满分做法有时很容易因为一些细节问题而挂掉，甚至分数不及朴素，以至于产生了“写正解太容易玩脱，不然到处写朴素骗分划算”这种技术性较弱，但实际效果很好的比赛策略。IOI这种能实时返回结果的赛制能一定程度减少这种问题。同时，IOI赛制也有一些原生优势，比如对开放性题目的支持；同时“不要求写完整程序，只要求写关键函数”的编程方式避免了硬盘读写的耗时，让选手不用再纠结于读入优化，也让一些特殊的题目（比如500万个询问的数据结构题）成为了可能。但同时，IOI赛制也有很多弊端，比如说一旦数据出错，后果将无可挽回；可以多次提交并看到结果让多种多样的骗分有了机会，因此给数据的质量提出了更高的要求；并且为了避免骗分数据不得不采用subtask制，导致得分拉不开，区分度降低；而且还可能出现类似今年IOI第一试时的评测机崩溃等各类问题，这些都对举办者的事先准备、应变能力提出了很高的要求。因此我觉得现在在NOI等关系到学生保送的比赛采用IOI赛制时机还不成熟，但可以考虑在WC、CTSC等比赛中采用IOI赛制，也可以采用相对折衷的方式，比如POI的ocen制（返回部分测试点的结果）之类。

在这次IOI旅程中，我开阔了眼界，学到了很多，同时也发现了自己的不足。这次IOI是我OI生涯的终点，但也是另一个起点。我会更加努力，完善自我，迎接新的挑战。