

IOI2014 中国国家集训队第一次作业

ACM/ICPC World Finals 试题解题报告

姓名：束欣凯 (SHUXK97)¹

目录

| | | |
|-----|--|---|
| 1 | ACM/ICPC World Finals 2011 H Mining Your Own Business..... | 2 |
| 1.1 | 题目简述..... | 2 |
| 1.2 | 10%的做法..... | 2 |
| 1.3 | 另外 10%的做法..... | 2 |
| 1.4 | 另外 20%的做法..... | 2 |
| 1.5 | 100%的做法..... | 3 |
| 2 | ACM/ICPC World Finals 2006 F Building a Clock..... | 3 |
| 2.1 | 题目简述..... | 3 |
| 2.2 | 简要题解..... | 4 |
| 2.3 | 测试数据的命制..... | 4 |
| 3 | ACM/ICPC World Finals 2001 I A Vexing Problem..... | 4 |
| 3.1 | 题目简述..... | 4 |
| 3.2 | 简要题解..... | 5 |
| 3.3 | 备注..... | 5 |
| 4 | 总结 | 5 |

¹ ID: SHUXK。联系方式: shuxinkai@126.com。

1 ACM/ICPC World Finals 2011 H Mining Your Own Business

1.1 题目简述

在一个连通图的结点中设置若干个“安全点”，使得无论哪个结点被切断，其它结点都有一条路径到达一个“安全点”。求最少要设置多少个“安全点”以及在满足最少的前提下有多少种方案。

边的数量 $N \leq 50000$ ，每个测试点的数据组数 $T \leq 20$ ，时限 2s。

1.2 10%的做法

10%的数据满足， $N \leq 15$ 。

此时可以直接暴力枚举安全点的设置情况，再在 $O(NM)$ 的时间内验证 (M 表示结点个数)。

时间复杂度为 $O(2^M M^2 T)$ (M 表示结点个数)。

1.3 另外 10%的做法

另外 10%的数据满足，每组数据都是一棵树。

对于这种情况，可以证明只在叶子结点设置“安全点”是最优的：

1. 必要性：每个叶子一定要设置“安全点”，否则如果它的父节点被切断，它无法连接到任何一个“安全点”；
2. 充分性：如果一个结点不是叶子那么它的度至少为 2，它至少有两条路径通向叶子。

所以无论哪个结点被切断都保证有一条通向“安全点”的路径。

这样就可以在 $O(N)$ 的时间内判断。

时间复杂度为 $O(NT)$ 。

1.4 另外 20%的做法

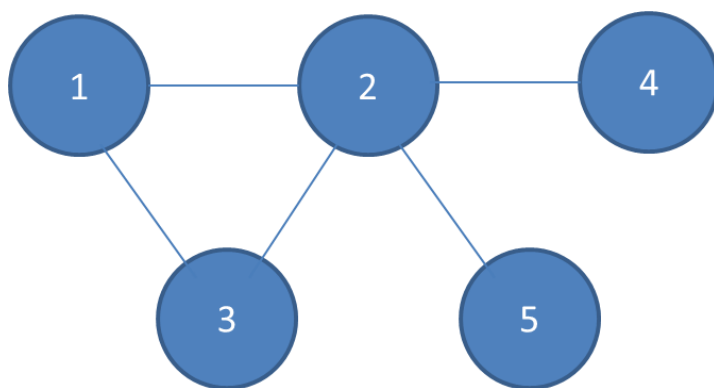
另外 20%的数据满足，每组数据中每个点最多在一个环中。

此时如果将环缩成点，那么整个图又可以看成一棵树。一个自然的想法就是在这棵新的树中只取叶子结点。但这样真的是对的吗？

对于原图中的单个结点，这样肯定是对的。但是对于环，就要再考虑一下。

对于叶子上的环，在它的内部肯定需要设置一个“安全点”；但是对于非叶子的环就一定不需要设置安全点吗？

答案是 No！具体反例如下页图：



虽然那个环缩点以后不是叶子节点，但是它也需要设置一个“安全点”。

可以看出，如果一个环上只有一个结点与外界连通，在这个环内部就需要一个“安全点”（但不能在与外界连通的那个结点上）。然后就是乘法原理计算方案数了。

特殊情况，如果整个图就是一个环，那么答案应该是 2 和 $\frac{N(N-1)}{2}$ 。

时间复杂度为 $O(NT)$ 。

1.5 100%的做法

50%的数据满足， $N \leq 500$ （这个纯粹是为了吐槽 HNOI2012 矿场搭建）。

100%的数据满足， $N \leq 50000$ 。

由上面的分析可以看出，这个问题和割点有很大的联系。因为如果一个割点被切断，那么它就会把图分成几个部分，那么这几个部分内都应该至少有一个“安全点”。那么，一个自然的想法是，去掉所有的割点以后有几个连通块就设置几个。但是这仍旧是错的，因为与两个以上割点相连的连通块并不需要设置出口。

然后只要求出只与一个割点相连的连通块的数量，方案数就可以按照乘法原理求出来。

特殊情况，如果图中没有割点，那么答案应该是 2 和 $\frac{M(M-1)}{2}$ （ M 表示结点个数）。

时间复杂度为 $O(NT)$ 。

2 ACM/ICPC World Finals 2006 F Building a Clock

2.1 题目简述

给定一个固定转速的转轴和若干个可用的齿轮（另外可以再用无限根转轴），求一种在转轴上安装齿轮组成时钟的方案。多解时选择使用转轴数量最少，使用齿轮数量最少，字典序最小的方案。每根转轴上最多有 3 个齿轮。

齿轮数 $N \leq 6$ ，每个测试点的数据组数 $T \leq 1000$ ，时限 2s。

2.2 简要题解

看到数据范围，这题就肯定是搜索了。齿轮这东西大家都很熟悉（胡渊鸣神犇的 2D 物理引擎里也早就加入齿轮了）。只不过要理解“每根转轴上最多有 3 个齿轮”这句话的意思。

稍微分析一下就可以知道，分针和时针是两个独立的系统，只有开头部分可能是公用的（这一点在题目中都提到了）。那么什么时候，一根转轴上会有 3 个齿轮呢？当且仅当一个齿轮驱动转轴，剩下的两个齿轮一个在时针的系统中，另一个在分针的系统中。

在搜索的时候，可以先不管时针，枚举构建分针系统的齿轮。每个齿轮只可能有两种状态：和前一个咬合，或者和前一个在同一个转轴上。如果成功构造出分针，再枚举两个系统的分界点，最后枚举剩下来的齿轮，看是否能构造出时针。

不过这道题需要非常注意细节，很容易写错。

2.3 测试数据的命制

这道题造数据比做题目本身要难一些。

纯随机基本上都 IMPOSSIBLE。为了让数据更有挑战性，我想出了两种方法：

1. 先构造出最终解的形态，再确定初始转速每个齿轮的齿数。有 50% 的概率将初始转速取反使其无解；
2. 将初始转速和 N 个齿轮的齿数都设成 1, 2, 3, 4, 6, 8, 9, 12, 16, 18, 24 中的一个（有正负）。这样可以保证约 50% 的有解率，并且有解的情况很随机，多解需要比较的情况也不少。

因为这道题的方法实在太弱，如果单组数据那么肯定所有人都 0ms AC，所以每个测试点有 1000 组数据，不过仍然没有到卡时的地步。

徐寅展神犇在做这道题时，发现了一个问题：在多解时即便有严格的限制，仍然有可能存在多解。对于 6 -9 H 24 0 18 X 1 E 12 D 8 C 4 这组数据，以下两种输出都是正确的：

Minutes: *C-OE-D-X 和 Minutes: *C-O-ED-X。

而我在命制数据时完全没有考虑到这个问题，后来才更改了数据。在这里对徐寅展神犇表示感谢并致以崇高的敬意。

3 ACM/ICPC World Finals 2001 I A Vexing Problem

3.1 题目简述

一个 $NR \times NC$ 的消方块的游戏。如果一个方块的左边或右边是空的，它就可以移动一格。移动后悬空的下落，下落后如果有两个或以上相同的方块相邻则可以消掉；都消掉后再下落，下落后可能再消掉……问最少移动多少步可以使所有方块消掉，并求方案。

数据保证存在 11 步以内的解法。

$NR, NC \leq 9$ ，每个测试点的数据组数 $T \leq 5$ ，时限 10s。

3.2 简要题解

对于搜索最少步数的问题，一般可以使用 BFS。但是直接 BFS 应该会超时。

首先应该去除重复的结点。由于本题的特殊性质，重复的结点很多（比如说把一个方块先向左移再向右移）。所以应该用 Hash 表去除这些重复的状态。这样程序就快了很多。

其次，由于步数最多只有 11 步，可以对当前局面的步数下界进行估计：

如果一个字母只出现一次，肯定无解；

如果一个字母只出现两次，消掉它们的步数至少为 $X_2 - X_1 - 1$ (X 表示横坐标)；

如果一个字母只出现三次，消掉它们的步数至少为 $(X_2 - X_1 - 1) + (X_3 - X_2 - 1)$ ；

如果一个字母只出现四次，消掉它们的步数至少为 $(X_2 - X_1 - 1) + (X_4 - X_3 - 1)$ ；

五次以上的探究就没有什么意义了。

加了这个优化以后，这个问题就可以很快解决了。

我的程序只用了这两个优化就 AC 了 UVA 上的原题，并且截至本文完成为止排名 UVA 上的第一名。但是我认为本题的优化远没有到达尽头 (A*, IDDFS 应该也不错)，希望各位同学踊跃 AC 此题，积极交流更好的优化。

3.3 备注

搜索本无定法，仁者见仁，智者见智。

我的方法我个人感觉也不是很优，我认为应该有更强的优化或者更好的方法。

为了让大家勇于尝试各种不同的方法，本题并没有出搜索量特别大的极限数据，不过比原始数据还是要强很多的（参见 UVA 上本题的 AC 记录）。

希望各位同学 A 掉此题，相互交流。优化无止境……

4 总结

这次我出的三道题有一道不算难的图论题和两道搜索题。由此可以看出 ACM 的题和 OI 还是有一定区别的。

接下来分别说一下这三道题：

那一道图论题是 HNOI2012 矿场搭建的**加强版**，这就很奇怪了。我很想吐槽 HNOI2012 抄了原题，还把数据出得更弱了。不过题目本身不难，不难想到正解。只要注意没有割点的特殊情况。出数据时要注意多种可能情况。我在数据中分了四种情况，每种各出五个数据。

虽然碰到了两道搜索题，但是这两道题并不像我想象的那么简单。首先，它们的题目描述特别长，我读懂和翻译题目就花了很长时间。其次，这两道题要注意细节。我写标程的时候就错了好几次才通过了 UVA 的数据，相当锻炼编程能力。第三，这两题的数据不好出。随机的数据几乎都无解。为了让数据有一定的挑战性，我只好考虑各种情况，对每种情况单独写程序，近乎手工地构造数据（虽然不知道最终是否会有挑战性，但是我已经尽力了）。

总而言之，这次做这三道题（或者说出这三道题？）对我来说是一次锻炼和考验，让我发现自己的不足之处。这三道题都不算难，看来集训队里的诸位神犇要狂虐我的弱题了。

接下来就是做其他人出的 ACM 题了。希望能在做题中提高自己的能力。