
目錄

简介	1.1
概述	1.2
打破常规	1.3
Petrus	1.3.1
Roux（桥式）	1.3.2
ZZ	1.3.3
CFOP（Fridrich）/FreeFOP	1.3.4
空槽F2L	1.3.5
Heise	1.3.6
怎样学习	1.3.7
如何进行还原	1.4
块构筑	1.4.1
找到好的构造	1.4.2
循环	1.4.3
插入	1.4.4
在现有解法中插入步骤	1.4.5
碰碰运气！	1.4.6
高级技巧	1.5
逆打乱	1.5.1
伪块，预打乱与NISS	1.5.2
反转NISS	1.5.3
实用公式	1.5.4
棱角对分析	1.5.5
如何练习	1.6
不要限制时间及模拟比赛	1.6.1
和高手比较（并学习他们的解法）	1.6.2
困难的打乱	1.6.3
针对性练习	1.6.4
快速打乱	1.6.5
学习！	1.6.6

一些其它方法	1.7
用EO开头	1.7.1
角先	1.7.2
比赛	1.8
如何写解法	1.8.1
备用解法	1.8.2
时间安排	1.8.3
其它资源	1.9

简介

本文章为 *FMC Tutorial ENG* 的中文译本，原文出处：http://fmcsolves.cubing.net/fmc_tutorial_ENG.odt

原文作者Sebastiano Tronto，WCA主页：[2011TRON02](#)。

翻译工作已获得原文作者授权。

授权邮件原文：

Hi,

yes, of course you can translate it! I'm happy to know that people are interested in FMC and I'd like my work to be accessible to more people. Here is the link to the text in .odt format (I've used Libreoffice to write it), in case you need it:

http://fmcsolves.cubing.net/fmc_tutorial_ENG.odt

I don't know about Gitbook, how does it work? Is it similar to github? Anyway, you can put it there if you want, as long as it is clear that I'm the author and you have translated it.

Good luck for your work :)

Sebastiano

翻译人员：蔡盛梁（石头球哥哥，[2014CAIS01](#)），廖梓跃（CuberL，[2013LIAO03](#)），Buglossoides。

由于时间仓促及能力有限，翻译工作难免出现疏漏，望谅解。

译注：本文章的“我”均指原文作者Sebastiano Tronto。

1. 概述

三阶魔方（应该说是任意一种魔方）竞速玩法很有趣，不过，尝试用尽可能少的步骤去还原是一种更有趣的玩法。而这就是最少步还原（Fewest Moves，或FMC，“Fewest Moves Challenge”）。

官方比赛

最少步是WCA（世界魔方协会，一个管理各类魔方项目比赛的组织）所承认的一个官方项目，官方关于最少步比赛的规则（[附则E](#)）可以总结为以下几点：

- 会拿到在一张写有打乱（打乱魔方的步骤序列）的纸，解法也要写在这张纸上的表格里。
- 时间限制：60分钟。
- 可以携带的物品：
 - 纸和笔（由裁判提供）
 - 最多三个魔方（自带）
 - 不限数量的贴纸（自带）
- 关于解法
 - 需要使用OBTM转动表示法来提交答卷：允许的步骤有整体转动（ $x,y2,[u],\dots$ ），单层转动（ $R,U2,L',\dots$ ）和双层转动（ $Rw2,Fw',\dots$ ），但不能使用内层转动（ M,E,S,\dots ）；在最终的成绩中，转体将不计入总步数，而其他步骤会被计为1步。
 - 解法的步数最多只能到80步（包括转体步骤）。
 - 不能与打乱序列有任何关联；除此之外，参赛者要能对他解法中的每一步做出解释。

迄今为止在比赛中最好的成绩是由日本选手Tomoaki Okayama（岡山友昭）所保持的20步的单次世界纪录和由德国选手Sébastien Aurox与美国选手Vincent Sheu共同保持的25步的平均世界纪录(三次取平均)。

非官方比赛及变种

许多论坛（包括speedsolving.com）和<http://www.ocf.berkeley.edu/~dadams/fmc> 每周都会举行线上非官方比赛。很明显没人能保证大家都遵守规则，因此这类比赛依赖于参赛者之间的相互信任。

最常见的变种比赛有“不限时比赛”，也就是不设任何时间限制（事实上限时一个星期就相当于“不限时”）以及“linear”，也就是不能回退或者消步，不能够寻找更多的解法，必须写下你第一个得出的解法（通常会把时间限制定在十分钟。）

本教程目标

写这个教程的目的是为那些能用于在最少步比赛中取得好成绩的最常见的方法做一个概述。有一部分我会给出很多的细节和实例，而有一些就只提供一个大体的解释并提供其他的资源以供日后学习。

资源及鸣谢

本教程中所提到的东西都来源于作者的经验，而这些经验并不仅仅通过练习来得到，也要归功于很多在网络上公开的资源，大部分资源可以在第八章“其它资源”部分找到，在教程中也会提到。所以在这里我要感谢世界各地的魔方社区，因为它们新的技术和方法才得以广泛传播，才能让每一个人可以轻松地学习到迄今为止的所有已有技术。我也希望我的教程能在这方面起到相同的作用。

原作者声明

你要知道，英语并不是我的第一语言，所以如果你觉得某些章节写的不好或者发现什么错误，请联系我：

sebastiano.tronto [at] gmail [dot] com

2. 打破常规

无论你使用的主要方法是什么，在最少步中限制自己只使用一种方法是一件错误的事。你应该突破自己使用单一方法的限制，尝试利用所有情况。

举个例子，假设你已经完成了一个2x2x3的块，现在你有很多的选择：你可以完成底十字的最后一个棱和F2L（CFOP），可以解决棱块色向（Petrus）或者尝试自由构筑更多的块（FreeFOP，Heise）等等。任意一种方法都可能做出好的解法，所以最好是把它们全都试一试（或至少多尝试一些）。

或许会有点矛盾，不过，打开思路及学习如何打破常规的最好方式，是学习许多“常规”，即学习许多还原方法。这里我将简要地介绍一些我认为在最少步中很有用的方法，并将不会涉及它们的发展历史或速拧中的利弊。每一个方法我都会给出在线教程的链接，但我建议你在Google或speedsolving.com上查找更多的资料。[这个教程](#)虽然主要是关于速拧，但也是一个了解4种最常用的方法（CFOP，桥式，ZZ和Petrus）的好的起始点。

Petrus

Petrus的步骤如下：

1. 构筑一个 $2 \times 2 \times 2$ 的块
2. 把 $2 \times 2 \times 2$ 扩展成 $2 \times 2 \times 3$
3. 棱块色向
4. 完成前两层（F2L）
5. 解决顶层（分为3个步骤）

要玩好最少步，良好的块构筑¹技巧必不可少，练习Petrus是一个不错的学习途径。想要学会如何有效解决第1、2个步骤，你每次都需要思考并尝试探索不同的方式去构筑块；重做最少步高手的解法也是很有效的学习方法。第一个步骤比较你的解法和求解器给出的优秀解法²之间的差距将对你大有益处，因为高手应该（几乎）总是能找到一个 $2 \times 2 \times 2$ 的优解。

第三个步骤教会你另一个重要的技巧——怎样辨认任意位置下的棱块色向。因为翻棱是在最少步中你会遇到的最糟糕的事情之一。这个步骤和第四个步骤，ZZ可能是比Petrus更好的老师。

你没有必要把顶层的所有公式都学了（其它方法也是）。下一章我会详细解释你应该怎样进行最少步。现在你只需知道“最后一层”这个步骤通常是不存在于最少步解法中就足够了。

¹. “块构筑”是一种把小块整合成大块的技巧，通常和CFOP（Fridrich）中的F2L的思路不大一样，但也可以把它当成是一种块构筑方式。相比较而言，一些其它的技巧，如棱块色向（Petrus，ZZ）、角先法的公式和循环更适合称为块构筑。↩

². 比如<http://laire.fi/jarcs/>，目前不是在线使用的。↩

Roux (桥式)

1. 构筑一个3x2x1的块
2. 在相对一侧构筑另一个3x2x1
3. CMLL (顶层角块, 无需保护M层)
4. LSE (最后六棱, 只需用到M和U层)

Petrus是学习块构筑的优秀方法, 但限制你自己只用Petrus是不对的: 学习Roux (特别是前两个步骤) 可以使其以某种其它方式完成。要学习这个方法也可以去研究高手们的解法 (现在最优秀的是Alexander Lau, 三阶速拧项目平均世界第二和2014年欧锦赛冠军。)

依靠最后一层公式, 第三步依然可以被保留, 然而第四步应该尽量舍弃 (至少不要使用标准的方法): 要知道, 每转动一次内层, 就会浪费两步。

[Waffle的Roux教程](#)。

ZZ

1. EOLine（解决所有棱块色向的同时还原DF棱和DB棱，剩下的部分可以只使用R、L和U层了完成）。
2. F2L
3. 最后一层

正如前面所提到的，学习ZZ是认识及解决EO这个实用技巧的一个最好的方法。首次接触EO这个概念可能会有点难，因为它比块构筑抽象得多，但不必惊慌，它不是不可能做到的！

第2个步骤或多或少有点像Petrus的第4个步骤，不过你需要同时构筑R层和L层的块³，这也可以提高你块构筑的能力。

[Conrad Rider](#)的教程。

³. 在速拧中一块一块构筑会比较好，但在最少步中不是这样的。↩

CFOP (Fridrich) /FreeFOP

传统的CFOP解法步骤如下

1. 完成十字(同一层的四个棱块)
2. 插入四个棱角对
3. OLL(最后一层色向)
4. PLL(最后一层位置)

传统的CFOP在最少步中不是一个好方法，不过在某些情况下，用不同方式插入棱角对是十分有用的。

在FreeFOP中，前两步被替代为自由构筑前两层。

不管怎样，我始终反对用OLL和PLL解决最后一层，除非你可以跳过这两步其中一步。

[Badmephisto的教程](#)

空槽F2L

空槽F2L不是一个可以真正还原魔方的方法，而是一个解决前两层的技巧。它被认为是一个介于“层先”和CFOP间的方法。步骤如下：

1. 完成十字
2. 第一层的三个角块归位
3. 使用空余的槽位插入中层的棱块
4. 如同CFOP或“层先”插入最后一个棱角对(F2L)

为了提高效率，你应该将前两步替换为块构筑并同时 will 将一些中层棱块归位。有一个变种是先完成十字和三个中层棱块，接着使用空槽完成三个底层的角块。

由于它的简便性，这个方法在最少步中是非常实用的。

Heise

1. 构筑4个2x2x1的“正方形”（有一个共同的颜色）
2. 整合这些“正方形”并调整棱块色向
3. 把2个角块和剩下的5个棱块归位
4. 最后3角（循环）

如果你决定不遵循高手的建议，只使用一种方法来最少步的话，Heise是个不错的选择，在linear玩法⁴中可以取得40步以下的平均。这是一个比较极端的方法，不过也很有效。前两步是构建F2L-1⁵和全部棱块色向的一个奇特而有效的方式。这里完全应用了“不要自我限制”和“利用不同情况”这两个概念，为还原块提供了更多的自由。

第三个步骤很复杂，不过它是一种还原两层再用公式解决顶层的有效替代方案。多多练习可以给你在有诸多块限制时的块构筑与块移动的能力（这也是这个步骤非常难的原因）。

最后一步你需要用到循环，它可以通过插入（这两个技巧都会在下一章解释）解决。

在[Ryan Heise的网站](#)上你不仅可以找到他的方法的详细解释，还有很多其它有用的信息（比如一些基本技巧）。

⁴. “linear”玩法不能够进行多次尝试，也不能消去或撤销步骤。↩

⁵. 我所说的F2L-1是指前两层减去一个棱角对。↩

怎样学习

显然，用上述方法提速不是你的目标。速拧可以提高你的观察，而且十分有趣，但我们不在乎速度。既然我们的目标是用尽可能少的步骤来还原魔方，你需要追求更高的效率。**color neutral**⁶也同样重要，它在处理一些“**Non Matching Blocks**”⁷时十分有用。

但是最少步和速拧的主要区别在于：在最少步中，你可以尝试不同可能性，比如在Petrus中，你在2x2x3后留下了6个情况不好的棱块，你可以尝试构筑不同的块或对你的解法稍加修改来改善你的情况⁸。

以下是上述方法的一些建议：

Petrus

- 当完成2x2x2后，你可以朝三个方向扩展。
- 尝试直接构造2x2x3来代替经过2x2x2的步骤。
- 在第4步中尝试使用伪块
- 在第4步中尝试改变最后一层的状态得到一个简单的case（甚至改变成Heise风格）。

Roux（桥式）

- 尝试同时构筑多个块。
- 尝试使用伪块。
- 完成第二个桥的时候改变CMLL的情况，在完成CMLL的时候改变最后6棱。

ZZ

- 完成EO后，做“Line”不是一个好主意，直接尝试块构筑。（不要影响到EO！）
- 棱块色向正确之后，你的魔方只需要通过旋转R、L、U和D层来还原：你可以用这些面来完成前两层。
- 和Petrus与Roux一样，你可以在完成前两层时尝试伪块以及改变最后一层状态。

CFOP / FreeFOP

- **FreeFOP**显然比**CFOP**要好；至少尝试做一个**Xcross**⁹。
- 尝试改变最后一层的EO，避免四棱翻转的情况；可以使用一些ZBF2L的公式，不过你不能死背公式，而应该理解它的工作原理。
- 一些理想的插入公式不是十分容易理解，(例如： $F2 U R U' R' F2$)：先学习，再理解。
- 尝试“**multislotting**”，即同时插入多个棱角对。最简单的情况就是把D层的转动作为setup，例如 $D R U R' D'$ 。你可以在[这里](#)，[这里](#)和[这里](#)找到一些公式，不过我建议你尝试

一些“自由”的方法，看我在[这个解法](#)中如何插入第二、第三和第四个棱角对。

6. 从任何一种颜色开始还原魔方，比如CFOP中的六色底或在Petrus从8个2x2x2中任何一个开始。↩

7. 有时候，特别是在最少步中，也叫伪块。它是一个在Roux、ZZ和Heise中十分有用的技巧，但它同时也可以在其他方法中使用。它是指在你原有方法基础上构造一个你不需要构造的块并可以放在同一个“槽位”。例如，在Roux中，第二个桥可以是四个中任何一个在第一个桥对面的3x2x1块。这个技巧如果和预打乱一起使用将会十分强大，将会在第三章做进一步介绍。↩

8. 这是一个良好的习惯，之后还会再说。↩

9. 同时完成十字和第一个棱角对。即可以一个2x2x2加上2个棱块。↩

3. 如何进行还原

引用Per Kristen Fredlund的话：

“可以（把最少步）看成这样：用两个步骤来还原一个魔方，第一步尽可能多且高效地还原大部分块（比如做一个好的构造¹⁰），第二步通常会通过往前面的步骤插入公式来使剩余的块归位¹¹。”¹²

这就是一般的做法，但并非每次都是这样：有时你可能会找到一个很短的解法，比如用块构筑完成前两层，然后用一个公式完成最后一层。还有一些其他的思路，会在第五章解释。

这个描述听起来太笼统了，这主要是因为没有办法具体描述：并没有一个标准的方法能让你每次都取得好成绩，你只能尽可能多的尝试多种策略，以不错失任何机会。

下面我会介绍最少步玩法中一些基础的技巧。你可能通过前面章节的学习和练习已经知道了一些，那你可以继续学习其他技巧。有一部分我会给出很多的细节和实例，而有一些就只提供一个大体的解释并提供其他的资源以供日后学习。

¹⁰. 一个部分解法，只剩下一小部分（2到6个）块尚未完成。↩

¹¹. 这是一个在已有解法中插入一些步骤来解决少部分块的技巧，后面将会有解释，耐心点！↩

¹². <http://www.speedsolving.com/forum/showthread.php?1566-Fewest-Moves-Tips-and-Techniques&p=16209&viewfull=1#post16209> ↩

块构筑

块构筑可能是最少步中最重要的技巧了。它是个很简单的概念，但想要精通则需要大量的练习。练习前面提到的以块构筑为基础的方法（Petrus、Roux、Heise和ZZ）是最直接有效的进步方式。

这里我会列出一系列可以用得上的基本技巧。首先就是Ryan Heise的网站，上面有一大堆的实例。快去找找！

调整与组合

基本技巧：构筑一个棱角对，这就是最简单的块了。首先你要把它们调整成可通过一步组合起来的状态，然后把它们组合起来。

一般来说，这个概念可用于构筑更大的块，例如把一个棱角对和一个2x2x1组合成一个3x2x1。

这些看起来没有什么，确实是这样。重要的是辨别出两个块已经对准，转动一步就能组合的情况，还有意识到某些步骤可以使2个块组合到一起。

这里有个实例：[13](#)

```
打乱：F U' D2 L D L2 D F R2 B U2 R2 D L2 D L2 D R2 U' L2 F2
```

有两个棱角对已经组合好了。如果你想把蓝红棱和蓝红黄角组合在一起，可以这么做：

```
调整：L2  
组合：U2
```

保护已完成的块

当你想构筑一些块的时候，有时会发现构筑的步骤会破坏其它的块，或者把块移动到不是我们希望的地方。避免这种问题的一种方法就是先把块移走使其不会被破坏。构筑完成后，有需要的话再把它们移回来。

下面是实例：

```
打乱：F R2 B D2 F D2 L2 B2 F' D2 F' L B U' F2 U2 L2 U B R2 F
```


做 $U^2 R^2 D R^2$ 后我们有了一个 $2 \times 2 \times 1$ 的块，可以通过 $F' D'$ 扩展成 $2 \times 2 \times 2$ ，不过 F' 会破坏红白棱和蓝红白角这一对我们希望保留下来的棱角对。我们可以做 U' （把它移开） $F' D'$ （组合）来保护它。

破坏与重组

解决块被破坏的另一种方式是使用“保护已完成的块”这一技巧先破坏，再重组。

下面给出实例：

打乱： $R U R' U'$

让我们忽略最后一层并假装不知道可以用 $U R U' R'$ 还原。现在只看前两层，可以看到 R' 可以把顶层的那组棱角对还原到F层，不过这样就破坏了前两层已完成的块了。这时，我们就可以使用“破坏与重组”了：

R' （破坏）
 F （保护已完成的块）
 R （重组）
 F' （把之前F移走的块移回来）

空槽

我们之前已经单独讨论过它，不过空槽也可以算是一种特别的块构筑方法。这个技巧可以利用未解决的部分来构筑块。通过一些空槽F2L的练习，我想不用什么例子你就可以理解了，不过我还是在这里放一个使用了空槽的实例：

实例¹⁴：

打乱： $U' R' L F' B U^2 R^2 B^2 L' B R D F^2 D^2 L^2 F^2 D' R^2 F^2$
 解法：
 $F' L'$ //一面差一角
 $F^2 L' B' L$ //空槽
 $F U' B U$ //空槽，实际上还插入了最后一个角块
 $F' R' B^2 R$ //空槽
 $F B L B L'$ //F2L
 B' //LL

一石二鸟

有时候一步可以完成两个块，或者说“完成两件事”。这里给个实例帮助理解。

实例：

打乱: D U' F2 U' R' F R2 B D' B R F B' U R' D2 L' R2 F2 B' U' B D B2 F2 U L F U' B2

这个解法来自Mirek Goljan和Guus Razoux Schultz（两人各自独立发现，没有合作）。这里可以4步构筑一个2x2x2的块：L U' F2 D'。这里的 F2 就是我们要说的：这一步在DF处构筑了一个2x2x1同时将橙绿块归位了，使得下一步可以构筑出2x2x2。

这种情况并不能强制出现，但还是了解了它比较好，万一不小心遇到了呢。

影响后续步骤

我们已经（简略）地提及完成F2L的方式对最后一层的影响¹⁵。这个想法也适用于块构筑：往往构筑一个非最佳的块¹⁶或者添加一些无关紧要的步骤使其有更好的后续解法、块构筑或者其它（比如EO）会更好。

举个例子：

打乱L2 D2 U R2 F2 D2 B2 U' R2 B2 U B U F D B2 U L D' R' F

做 L2 R B 可以很简单地得到一个2x2x1块，但只要加上一步就可以变成2个：L2 B R B。

注意EO¹⁷！

许多人可能在研究不同的方法时常会碰到EO这一步。前面也说过，有越多色向错误的棱块就会越难。在最后才EO并不高效。如ZZ在一开始就把它解决了会很方便，但也限制了块构筑。根据很多人（也包括我自己）的经验，在块构筑的过程中解决EO是最好的。有ZZ和Petrus经验的话可以很容易看出需要几步可以解决棱块色向。如果你还没有这个能力，记住最少步中你每次都可以回退和修改你的解法，当EO有点麻烦的时候，不妨回退修改几步，或许情况会变得好一些。

我需要构筑哪些块？

最少步的黄金准则是善于利用不同的情况：构筑一个2x2x2、3x2x1、两个2x2x1或者其他很多种类的块都可能成为一个好开头。尝试所有的可能性。

两种合理的方法是：

1. 尝试完成一些大一点的块，比如2x2x3或者F2L-1。
2. 通过一些小步骤完成一堆小块，最后把它们组合起来。

Erik Jernqvist提出以下作为好开头的步数要求¹⁸：

2x2x1+一个棱角对：3

2x2x2：4

两个2x2x1：5

2x2x3：9

F2L-1（TEC）：14

F2L（233）：17

就个人而言，我觉得这些估算都是合理的，特别是前三个。不过要记住，一个好的开头取决于有没有好的后续解法。如果你12步完成了F2L-1但有4个棱色向错误，其中一个还是F2L棱（原位翻转），你一定会舍弃它；反过来讲，12步2x2x3但棱块色向都对了会是一个更好的选择。当然EO并不是你唯一要考虑的因素，但却是最重要的因素之一。

另一个原则是：不要¹⁹只顾着还原F2L而不去理会最后一层。原因很简单：不管你会多少公式，最后一层很糟糕的话也会用很多步，而完成F2L后还原魔方的自由度就大大降低了；与此相反，F2L-1是一个不错的分步目标，因为它给你的自由度还是比较高的，尽管大部分已经完成了。

现成的块：如何处理？

有时候打乱完你会发现很多块已经完成了，或者第一步无意中构筑了多余的块（通常是棱角对）。这种情况下，最好是把它们利用起来，而不是无视它们我行我素。怎么利用呢？大致有以下三种方式：

1. 很显而易见，首先扩展/匹配已完成块。
2. 扩展/匹配已完成块，但同时关注其他部分并尝试构筑其它的块。通常这是最好的做法。
3. 不扩展已完成块，而是构筑新块，尽可能把其它块保留下来。

此外，理解什么时候一个块没有保留价值而应该毁掉它去构筑新块，是很重要但也非常难的事。就个人而言，我非常难以做到“放弃”和“扔掉”一个块，也意识到这经常导致一些麻烦，特别是在时间安排上。

更快得出好解法的诀窍

“快”并不只是在速拧中提及，在“更快找到好的块构筑开头²⁰”。更快找到好开头是很重要的，因为能节省时间（一个小时并不长！），你要尝试，或至少留意到所有可用的开头。

最简单的块是棱角对，或称为 $2 \times 1 \times 1$ 块。它可能在打乱后就出现了。如果有的话，你可以用上一节“现成的块：如何处理？”中的方法来解决它；如果一个都没有，你应该学会识别那些可以一步组合起来的棱角对（参照“调整与组合”）。如果你没法很快地识别，或者想减少思考²¹，可以使用“暴力”法：从U、U²、U'开始尝试所有可能的转动，然后是R、R²、R'，以此类推，每次转动都检查一下有没有构筑出棱角对。

更有用但也需要花更大的精力的高级技巧，是检查每一个 $2 \times 2 \times 2$ （一共有8个）。你可以这么做：检查每个角²²及能和它组合的三个棱，然后把它们构筑成 $2 \times 2 \times 2$ 。试着不要做任何“尝试的”步骤，这样你就可以不用重新打乱便检查完所有的角。通常我看到很糟糕的 $2 \times 2 \times 2$ （指需要很多步骤）我会直接忽略它。这个技巧大多数时候可以让你找到一个不错的 $2 \times 2 \times 2$ （一般是个好开头）。

我建议你不转动而是“观察”构筑 $2 \times 2 \times 2$ 的每一步的另一个原因是，要更快更好地进行块构筑，能够不转动而“计算”²³块的移动是很有用的。我们之前讲过的Alexander Lau，它能够在速拧前的15秒观察时间内计算出整个 $3 \times 2 \times 1$ 块（Roux的第一步）。他的规划很快速准确，这使得他可以在还原这个块的同时预判²⁴并（部分）规划第二个块。

下面这个游戏训练你的规划能力：找一个朋友用3步打乱你的魔方，然后你试着用3步还原²⁵（这很容易）。然后尝试4步等等。你可能会发现到了6、7或者8步时变得很困难，这取决于你的水平。当你到了9或者10步时也毫无压力时，恭喜你！

¹³. 从这里开始，当我给出打乱并解释（完整或部分）解法，我可能会提及颜色。这时默认魔方是西方标准配色（BOY, Blue-Orange-Yellow）并以白顶绿前放置。↩

¹⁴. 这个解法由Edoardo Disarò的解法稍加修改而来的，使其不涉及之后要解释的技巧。↩

¹⁵. 比较出名的是ZBLS（有时会被误称为ZBF2L）和Winter Variation，其实还有很多：自己找找！↩

¹⁶. “非最佳”是指一个（完整或者部分）解法步数不是最少的。↩

¹⁷. 棱块色向（Edges Orientation）。↩

¹⁸. 这个列表来自speedsolving.com。很明显，怎样算“好开头”取决于你的水平。给出的步数是以一个高手所期望的好成绩为基础估计的。如果你还不怎么擅长的话可以适当放低目标，不要为了找好开头而浪费太多时间。↩

¹⁹. 当然，总是会有例外。↩

²⁰. “开头”可以是 $2 \times 2 \times 2$ 、 $3 \times 2 \times 1$ 或一些小的块，在一些情况下还可以用 $2 \times 2 \times 3$ ，大体而言2到7步。↩

²¹. 因为你需要持续一个小时的思考，避免受到影响是一个好习惯，除非你很懒。↩

²². 我从ULB开始，顺时针一圈，然后做个 $\times 2$ ，重复一遍。↩

23. “计算”这个词意思和下棋差不多：玩家说能够计算6、7或8步意味着他可以思考出6、7或8步的步骤或逆步骤。↩

24. 在速拧中，“预判”是指在进行一个步骤时思考下一个步骤。↩

25. 你可以在HTM、QTM和STM之间进行选择，但首先要确定使用哪种计数方式！↩

找到好的构造

一旦通过块构筑得到一些好的情况后（比如F2L-1），就很难再利用前面讲过的技术来继续前进。你的目标是找到一个叫做“skeleton”（译注：后面将译作构造或者构造序列）的东西，它是指将魔方还原到只剩下少量几个块未解决的状态。最好的情况是只剩下三个角块未解决，这三个角块将组成三循环²⁶。剩下四个角块，五个角块或者三个棱块也是不错的情况，主要还是要看你前面的构造所用的步数而定。

如果我们发现用块构筑技巧不得不破坏已经做好的块，我们该怎么做？Heise法是一个不错的出发点。

Heise法的**第三步**并不仅仅要求完成到F2L-1,而且还要全部的棱块色向正确。如果这些棱块不正确，你可以：

1. 现在修正它们。通常不推荐这种做法，除非只用很短的步骤就可以修正。
2. 改变前面的步骤，让所有的棱块最终是正确的。
3. 等到构造完再修正它们。

总之，找到一个好的构造的途径有很多，最好的练习方法和习惯是看一些线上的实例（比如<http://fmcsolves.cubing.net/>）来研究其他最少步玩家的解法。

你还可以专门练习这个步骤：使用qqtimer，然后选择last slot + last layer的打乱方式。

当然你也不一定非得在构造之前完成到F2L-1，但通常是最简单的做法²⁷。无论如何，试着保留那些碰巧遇到的组成最后一层的小块（2x2x1块或者是棱角对）。

²⁶ 三循环是三个块组成一个循环，例如A-Perm和U-Perm均为三循环，以及OLL `L F R`
`F' L' F R' F'`。↩

²⁷ 你可以不这么做，不过这会使不确定性增加，需要随机应变。↩

循环

根据[speedsolving's definition](#)，在魔方中²⁸，空穴（Commutators，为了便于理解，后文均译为循环）是一类类似这样的步骤：

$$A B A' B'$$

在这里面，**A**和**B**是步骤序列而**X'**则是**X**²⁹的反序列。像这样的循环，可以缩写为：

$$[A, B]$$

通常，在实践中，“循环”指的是“三循环”，我们在这里也使用这种概念。

相比于块构筑可以解决大部分块但会很大程度影响魔方的其他部分，循环解决少量的块（通常是三块）但保留了其余的部分。因此，块构筑、插入（我们将在下面的章节介绍）和循环都是得到一个好的最少步解法的基础。

角块循环

角块循环是最少步玩法中最有用的技巧。 $R U' L' U R' U' L U$ ， $R^2 B^2 R F R' B^2 R F' R$ 和其它你所知道的公式都可以当作循环来使用。

至于如何学习使用循环（在教程中我不会解释），我建议你去看一下[Brian Yu的教程](#)，里面包括了文字和视频，做得非常好。

在最少步中你只需要了解“纯粹”的八步循环（译注：**pure**，即不需要**setup**的，特指角块循环。后文会直接译作八步循环）。如果你愿意，你可以看看九步循环和其他情况，不过当我们讲到插入的时候你会看到的，你几乎不会再传统最少步中使用它们。³⁰

棱块循环

一旦你学会了角块循环后，你会发现棱块循环很难理解，特别是像这样的：

$$[U R U', M'] = U R U' M' U R' U' M$$

可惜的是，像这样的循环并不是很经常用到（即使有时很有用），因为它使用了**M**层的转动，而这样的转动在解法中算作两步。

每个人都该知道的循环是：

$$[M', U2] = M' U2 M U2$$

也可以加上一些setup的步骤³¹：

$$[U:[M', U2]] = U M' U2 M U2 U' = U M' U2 M U$$

要记住，在官方比赛中你不可以写上转动内层的步骤，所以循环 $[M', U2]$ 应该记成：

$$M' U2 M U2 = R' L \times U2 R L' \times' U2 = R' L F2 R L' U2$$

需要注意的是，前两步 $R' L$ 可以被交换。

另外你要注意到前两步并不真正的影响参与循环的三个棱块， $R' L F2 R L' U2$ 等效于 $L F2 R L' U2 R'$ 或者 $F2 R L' U2 R' L$ ，或者我们可以颠倒 R' 和 L ，变成 $R' F2 R L' U2 L$ 。

这些在你想消去一些步骤的时候尤为有用，它可以让循环的第一步（或者一些步骤）跟前面的解法的逆序相契合（或者后面的一些步骤与后面的解法的逆序相契合）³²。

还有一些棱块三循环也不需要用到内层转动，比较常见的如下：

$$\begin{aligned} R2 B2 L2 U B2 R2 F2 D \\ R2 B2 L2 U L2 B2 R2 D \end{aligned}$$

注意前两步并不会影响我们所要解决的三个棱块，所以我们可以“改变”它，做法类似我们前面对 $R' L F2 R L' U2$ 的改变。

对于棱块循环最后再提一点，就算没有三循环，也可以利用序列 $M2 U2 M2 U2$ 和 $R2 U2 R2 U2$ ，都可以交换两组棱块。

块循环

如果你很认真的阅览过Ryan Heise的网站，你应该已经知道了一些关于“组三循环”或者是块循环的知识。如果你已经了解了角循环，你会发现块循环是很直观的，例如：

$$[L Dw L', U'] = L Dw L' U' L Dw' L' U$$

这个方法在Heise法的第三步中很有用，也经常用于同时完成一个角块三循环和一个棱块三循环，比如最后一层公式 $M F U F' U' F' L F R'$ 可以被看作：

$$[R:[L' Dw L, U']] = R L' Dw L U' L' Dw' L U R'$$

其实就是带了一步setup的块循环。

J-Perm 也可以被看做是一个组三循环：

打乱：

$$R\ U^2\ R'\ U'\ R\ U^2\ L'\ U\ R'\ U'\ L\ U$$

解法：

$$[R2:[Fw,D\ B2\ D']] = R2\ Fw^2\ D\ B2\ D'\ Fw^2\ D\ B2\ D'\ B2$$

28. 这个定义来自于数学上的群论。↩

29. 例如 $U\ R$ 的反序列是 $R'\ U'$ 而不是 $R\ U$! ↩

30. 除了这些例外 ↩

31. $[A:B]$ 这种记法表示共轭，即 $A\ B\ A'$ ， A 通常被称为setup步骤。↩

32. 比方说我们的F2L以 $U\ R^2\ F'$ 结束，然后我们使用公式 $F\ R\ U\ R'\ U'\ F'$ ，就可以消去三步：
 $U\ (R^2\ F\ F'\ R)\ U\ R'\ U'\ F' = U\ R'\ U\ R'\ U'\ F'$ ↩

插入

多次在本教程中提起它之后，是时候真正讲解插入技巧了。

当我们以某种方式得到一个好的构造后，我们假设只需要一个角块三循环就可以完成它。我们该做什么？我们显然可以直接做一个角块循环，但是，如果你学习了所有的角块三循环情况，你会知道一个角块循环可以多达12步。而最好的情形却只需要八步，多出了这四步就不好了。但有一个几乎可以在八步内就完成一个角块三循环的方法：插入。

简单的插入

插入的原理并不难理解：如果只剩下三个角块就能复原，我可以一步一步看我的构造步骤，并在能用八步就解决这三个角块的情况下解决它们。这个循环并不会影响除了这三个角块之外的其他部分，而余下的步骤会继续解决其他的部分，而这三个角块也会归位，因为我们已经用插入的循环中解决了它们。

这就是怎么用八步来解决一个角块三循环的方法。这个技巧如何进阶？在尝试过所有可能的插入之后，我们当然会选择一个可以消去最多步骤的插入；通常来说，检查出所有八步循环，并从中选择出一个最好的就足够了。但在[及其罕见的情况下](#)，最好的插入是一个九步（或者更长）的循环，但这种情况不太可能发生，所以并不是很有必要去尝试每一种循环。

为了可以更方便的在检查构造步骤时跟踪这三个角块，很多人建议用白色贴纸³³来贴魔方，并标上1,2,3（你喜欢的话也可以用字母A,B,C）³⁴。而我的做法是，带一个比较便宜的而且贴纸亮度不高的魔方，用铅笔写在上面。

下面给出一个例子³⁵：

打乱：D B2 U' F2 L2 D2 R2 U F2 U2 L2 R' D2 B L' U' R2 F2 R B F2

构造：

B' U' D L' F' //E0 + blocks

D2 L2 D' L //Pseudo 2x2x3

U2 R2 U' R' //Pseudo 2x2x1

U L' U R' U' L U2 R' L' //All but 3 corners

在这个时候，取一根铅笔然后在蓝红黄角块上的红色面写上“1”，在蓝黄橙角块上的橙色面写上“2”，在橙蓝白角块上的白色面写上“3”³⁶。然后使用公式 $L B2 L F L' B2 L F' L2$ 还原，并重新打乱它。

现在我们可以解决这三个角块，但我们需要九步 $R2 F R B2 R' F' R B2 R$ 。所以我们转动构造步骤的第一步 B'，然后看我们是不是能得到一个更好的情形：看起来不是，依然需要九步。继续下一个步骤 U' ³⁷：我们知道现在可以用八步循环来解决这三个角块了 $L2 F R F' L2 F$

$R' F'$ ！如果我们使用这个插入点，最终的解法会是这样：

$$B' U' (L2 F R F' L2 F R' F') D L' F' D2 L2 D' L U2 R2 U' R' U L' U R' U' L U2 R' L'$$

但是，我们还有更好的选择。继续下一步 D 会发现这个时候需要插入九步的循环³⁸。再接下去检查一些步骤，直到 $L' F' D2$ 。现在我们可以使用八步循环来解决这三个角块 $D' F2 D B2$ $D' F2 D B2$ ³⁹。但会发现不止是这样：最后一步与这个循环的第一步抵消了！如果我们想用这个插入点来还原，解法如下：

$$B' U' D L' F' D2 (D' F2 D B2 D' F2 D B2) L2 D' L U2 R2 U' R' U L' U R' U' L U2 R' L'$$

等价于

$$B' U' D L' F' (D F2 D B2 D' F2 D B2) L2 D' L U2 R2 U' R' U L' U R' U' L U2 R' L'$$

消去一步，耶！所以我们只用了七步就解决了这三个角块。

出于更周全的考虑，我们需要继续寻求更好的插入点直到这个构造步骤被检查完。事实上，最好的插入出现在后面：

$$B' U' D L' F' D2 L2 D' L U2 R2 U' R' U L' * U R' U' L U2 R' L'$$

$$* = L F' L' B L F L' B'$$

得出解法：

$$B' U' D L' F' D2 L2 D' L U2 R2 U' R' U L' L F' L' B L F L' B' U R' U' L U2 R' L'$$

其中 L 和 L' 抵消，所以得出：

$$B' U' D L' F' D2 L2 D' L U2 R2 U' R' U F' L' B L F L' B' U R' U' L U2 R' L'$$

这里不再详述关于棱块的插入，但你应该掌握棱块三循环和棱块两两交换的方法，使用公式 $M2 U2 M2$ 或者 $R2 U2 R2 U2 R2 U2$ 或者其他变形公式⁴⁰。

多个插入：分开的循环（棱块三循环和角块三循环）

构造后的魔方并不总是留下一个三循环，插入也可以用于解决更多的循环。

正如我们所看过的，由一个角块循环和一个棱块循环组成的两个三循环，可以利用组循环（可能包括setup步骤）来解决。另一种思路是利用"Sune"⁴¹或者其变式来解决棱块，于是我们需要循环⁴²的只是受影响的角块。这些方法都应该牢记于心，但并不是很经常用到⁴³。最标准的解法是插入两个循环。

用数字标记了角块和棱块之后⁴⁴，就跟做简单的插入一样一步一步检查，但要同时注意角块和棱块以及能否使用组循环和"Sune"。完成之后，就可以把你带有两个循环的最终解法写下来。你还可以做另一种尝试：如果你想保留找到的角块循环，并找到更好的棱块循环，你可以写下只使用了角块循环的解法，然后你就会得到一个比原来的稍长一点，只剩下三个棱块的新的构造。现在你可以用简单的插入来还原它。为什么我们要多做一次搜索？这是因为循环的内部也可能是很好的插入点。你也可以先找出棱块循环，然后再找角块循环。这里有个好的解法，尽管理解起来有点困难。

你完全可以利用这个方法来解决需要两个甚至更多分开循环（三个循环或者是两两交换）的情况，但会显得比较复杂。

多个插入：两个或三个需要翻色的角块

当你需要修正两个角块的色向时，你可以试着在某些点插入 $[F L' D2 L F', U2]$ （12步），而对于三个角块的情况，你可以使用 $U' B U' F U2 B2 D' R2 U D F' U' B$ （13步）。但这些通常不是最好的解决思路，特别是第二种情况。

解决两个或者三个需要翻色的角块一般是插入两个角块循环。第一个循环只需要把这三个需要翻色的角块（对于两个角块的情况，是两个需要翻色的角块和任选一个角块）加入循环，并尽可能多的消去步数。然后在插入后得到的新构造中插入第二个循环。

使用这种方法时，只要在需要翻色的角块上面画个X或者其他什么符号；而如果你打算使用前面所提到的翻角公式，我建议你在上面画一个箭头，以便提醒你怎么翻转这些角块（是顺时针还是逆时针）。

找到第一个循环后，擦掉这些标记，然后在贴纸上写下1,2,3。

你也可以利用这种方法解决两个需要翻转的棱块，但我建议你选择放弃，因为棱块循环通常需要较多的步数。

多个插入：四个角块

如果魔方剩下四个角块，最坏的情况是四个角块都在对的位置但色向错误，这种情况需要插入三个循环，应该选择放弃。⁴⁵

除去这种情况，还有其他三种情况：

1. 一个角块需要翻色，其他三个角块组成一个色向错误的三循环⁴⁶。
2. 有两对角块需要两两互换，且色向正确⁴⁷。
3. 有两对角块需要两两交换，且色向不正确。

上面所述的情况都可以用两个循环来解决：第一种循环是需要先完成四个角块中的一个，而无需顾及⁴⁸另外三个，然后第二个循环就需要在第一个插入后的新构造步骤中插入，完成其余三个角块。

对于上面三种情况，我用下列三种方法来标记魔方：

1. 用 **x** 来标记那个需要翻色的角块（我并不关心它应该往哪个地方翻转），然后用数字1到3来标记其余三个角块；而这个时候这个循环的“色向”是错误的，1对应2，2对应3，3却不会对应1，而是这个角块的另一个面⁴⁹。没事，给这个面标上数字4。
2. 给其中一组角块标上两个**X**，给另一组角块标上两个**A**。
3. 我们用了四个数字来标记一个“色向”错误的三循环，而对于一个“色向”错误的二循环，我们需要三个数字：与情况1同理，给这两个二循环分别标上1到3和**A**到**C**。

对于第一种情况，第一个循环可以是**X**到3,3到4,4到**X**。但1->2->3->1这样的循环就不可取，这会导致最终留下两个错误的角块，而非三循环。

同样地，我建议不要用相同的方法来处理四个棱块的情况。

这里有一个改编过⁵⁰的**优秀实例**：由João Pedro Batista Ribeiro Costa保持的南美洲记录。

打乱：L U2 D' L' U2 B' D B' L U2 F2 R2 F' R2 L2 F' U2 D2 F

解法：

F D' # U2 * R //E0

D' F' L2 //2x2x2

F2 D F2 //Pseudo F2L-2

B' D B //F2L-1

F' D' F' //AB4C

* = U R D' R' U' R D R' //First commutator, 5 moves cancel

= L' U' R' U L U' R U //Second commutator, 3 moves cancel

最终解法：F D' L' U' R' U L U' R2 D' R' U' R F' L2 F2 D F2 B' D B F' D' F'

多个插入：五个角块

在所有剩下五个角块的情况中，只有一种情况可以构成五循环，从而只需要插入两个三循环就可以完成。其余大部分情况需要三个循环。而当五个角块都在正确的位置而色向错误，则需要插入四个循环。这里不讨论需要三步或者三步以上的情况而只讨论需要两个循环的情况（即使你有时确实要尝试插入三个循环）。

最简单，也许也是最有效的方法是使用二步搜索法，就像剩四个角的做法一样。给角块标上数字1到5后，你重新一步一步检查构造序列，然后找到任何可以解决序号连续的三个角的循环，这些循环如下：

```
1 → 2 → 3 → 1
2 → 3 → 4 → 2
3 → 4 → 5 → 3
4 → 5 → 1 → 4
5 → 1 → 2 → 5
```

这显然会比只找一个循环花费更多的时间，同样地，找到八步循环就足够了，而你每找到一个就把它写下来。

当你完成第一个步骤后，你就选择你找到的循环当中最好的（消去步数最多的）一个⁵¹，然后插入这个循环。现在你就有了一个新的，只剩下三个角块的构造序列，然后你应该明白要做什么了。

这个方法并不能保证你所得到的就是最优解法：为了安全起见，你应该检查（至少是大部分）第一步所找到的循环，然后完成很多的第二步（也许不起作用）。这种方法会浪费很多时间，但可能得出更好的解法。

如果你不清楚还能不能再得到更好的解，记住完成一个角块五循环的目标是10到11步。

还有一个更快，而且可以一步到位，但稍微有点复杂的方法。它几乎和二步搜索法中第一步的做法相同，但要同时记下找到的循环和还原的角块。这个时候，你甚至不用再去碰魔方，就可以知道选择哪对循环来完成角块五循环。

要理解这个方法，我们需要一些关于循环的理论⁵²；不用担心，我会尽量将它缩减。

首先，按照平常的记法，循环记为：

```
(1 2 3 4 5)
```

意思是角块1指向角块2,角块2指向角块3...角块5指向角块1。在这种记法中，循环(2 3 4 5 1), (3 4 5 1 2)等等都等价于上面那个循环。我们要做的就是把这个五循环拆分成两个三循环，例如：

```
(1 2 3)(3 4 5)
```

但这样的循环其实是行不通的。如果你想知道为什么行不通，你可以阅读脚注上面链接的文章，或者你也可以自己动手尝试一下！总之，真正可用的分解应该是这样的：

```
(1 2 3) (4 5 1)
(2 3 4) (5 1 2)
(3 4 5) (1 2 3)
(4 5 1) (2 3 4)
(5 1 2) (3 4 5)
```

所以，第一个循环的第一个数字和第二个循环的最后一个数字应该是相同的。

注意，顺序很重要。可以看到，(1,2,3)可以出现在(4,5,1)前面，也可以出现在(3,4,5)的后面。也就是说，当你找到一个好的循环，比方说，循环(1,2,3)，你就可以选择在这个插入点后面的步骤中插入一个循环(4,5,1)，或者在这个插入点前面插入一个循环(3,4,5)。同理，可以尝试上面列举的所有循环。

即便这个方法会比上面的方法更快一些，但却不允许你在一个循环的步骤中插入另一个循环，因此，我建议你在时间不充裕的情况下使用这个方法，或者只作为“初步分析”，看看用这种方法能得到多少步的解，然后再进行第二步来寻求更优的解。

多个插入：五个棱块

通常剩下五个棱块的情况最好放弃，但如果你打算进行尝试，你可以使用跟角块相同的方法。

但有一种情况是可以在很短的步骤内解决的：你可以使用六步五循环：

```
M' U M U'
```

即使是再加上一些setup步骤，也是很不错的。试着了解哪些情况可以用它以及它的一些变式来解决，比如它的变化公式 $L F L' R U' R$ 。如果你得到一个剩下五个棱块的构造，你可以给它标上数字然后你可以给它标上数字然后快速转动看看能不能使用这些公式，但不要在这个方法上投入太多时间。

其他插入：两个角块和两个棱块

有时你会找到一个只剩下两个角块和两个棱块的构造（比如J-Perm,T-Perm,V-Perm等PLL），

在这种情况下，了解一些10步的公式是很有用的：

```
Fw2 R D R' Fw2 R D' R D R2 (J-Perm)
Rw' U Rw' U2 R B' R' U2 Rw2 B' (T-Perm + 翻角)
```

还有一些11步的公式⁵³，比如：


```
R U2 R' U' R U2 L' U R' U' L (J perm)
R2 D B2 D' Fw2 D B2 D' Fw2 R2 U' (J perm)
R2 Uw R2 Uw' R2 F2 Uw' F2 Uw F2 U' (T perm)
R' U R U2 L' R' U R U' L U2 (J perm + 翻角)
```

除了这些公式之外（事实上也不只有这些公式），还有它们的逆公式和镜像公式也可以同时还原两个棱块和两个角块。

不过要注意这些公式的逆公式其实是相同的情形，只是让你在不用学习新公式的情况下增大消步的几率。

不要指望可以消去很多步骤，但你掌握越多公式，就越有可能得到优秀的解法。

这里有个不错的[实例](#)，不过使用了NISS技巧，这个会在后文解释。

其他插入：共轭与还原

还有一种情况是，你可以只用一个插入来解决分别构成四循环的四个棱块和四个角块。

你可以用如下的方法来解决这种情况：把这八个未复原的块放在同一层（**setup**），并且要可以用一步就解决这两个四循环，然后就转动这一步，然后再把这些块都放回去（取消**setup**）。相同的方法也可以用于八个块分成四个二循环的情况，在这种情况下，这个转换的步骤将会是 180° （比如U2）⁵⁴。

这里有个很好的[例子](#)，我找到这个构造后，插入了三个循环。但Mirek Goljan 给出了一个更有趣的插入，就是使用了这个方法：

```
Scramble: R2 L2 D2 F2 D' R2 U' B2 D' F2 U2 F' D2 L' F U B F2 U2 F2 L
Solution: U2 F B' L2 D2 F' U F2 L' B2 L F2 L' B' L' D L B' D L U L' D' L U2 R F2 R' L2
B
Skeleton: U2 F B' L2 D2 F' * R F2 R' L2 B
Insertion: * = (B D R2 B R' B2 D U2 F') U (F U2 D' B2 R B' R2 D' B')
```

有一些LL公式其实就是利用了这个方法：比如 $(R B2 R2 U2 R) B (R' U2 R2 B2 R')$ 以及其他的一些[公式](#)。

步数的估算

这里我给出几种常用的插入大致所需要的步数，这只是一个大致的估算，而没有经过严谨的证明，而且这些数字也取决于以下几点：

- 你了解多少循环/公式和你观察它们的能力。
- 构造的长度：长的构造序列会有更多的插入点，所以会有更多机会找到消步（当然也不要因为这个就去选择一个长的构造序列！）。

这些估算的数字可以让你知道到底有没有必要花时间去找插入：如果你的目标是找到一个30步之内的解，而你有一个23步的构造序列而且剩下三个角块，你就很可能成功；但如果你的构造序列超过25步那你可能就需要一些运气了。

你也可以利用这个来比较不同的构造序列：18步剩下四个角块的构造序列就可能比25步剩下三个角块的构造序列要好一些。

下面列举这些数值⁵⁵：

- 角块三循环：6步。
- 棱块三循环：7步（随机性很大）。
- 两个角块翻色，两个循环：8步。
- 三个角块翻色，两个循环：9步。
- 四个角块，两个循环：10步。
- 角块五循环，两个循环：11步。
- 两个棱块和两个角块，两两交换：10步。

找插入工具

由董百强开发的**找插入工具**（IF，Insertion Finder），是一个找插入、验证你所找到的插入是不是最优的利器，这个工具最多可以找到4个插入。

在一些简单的情形下（三个角块或者三个棱块）它很有用，但在一些复杂的情形下会给出一些人类不可能得到的解：要有选择的利用它。

33. 这就是为什么允许你在比赛中带数量不限的贴纸。↩

34. 贴纸要以1到2、2到3、3到1的方式贴在魔方上以便还原。↩

35. Pseudo，伪块，后文讲解。↩

36. 有很多等效的贴法，你可以从任意一个角块的任意一个面开始，只要保证它们是连续的。↩

37. 注意:当一个构造有两个相对层的步骤时,试着交换它们的位置,说不定能找到一个好的插入,不一定会有,但你也不知道它会不会就出现了。↩

38. 最后两步 $U' D$ 等效于中层转动 E' ，所以并不会影响角块：所以在这种步骤的前面和后面，我们的三循环是相同的，只是相当于做了一个整体转动（这个情况下是 y'/y ）。↩

39. 或者 $B^2 U' F' U B^2 U' F U$ ↩

40. 也可以做两个循环，跟插入两个角块循环同理。↩

41. $R U R' U R U^2 R'$ ↩

42

42. 还需要继续还原，通常会增加一个插入。↩
43. 除非你用了很多setup步骤，但这会导致解法变得很长，效率变得很低。↩
44. 我更喜欢在角块上面标数字，在棱块上面标字母，这样不会混淆。↩
45. Speedsolving.com论坛上FMC thread有展开相关的讨论，Sébastien Auroux（一位非常权威的玩家）的意见略有不同。但四个色向错误的角块依然是剩下四个角块中最差的情况，除去单纯的步数因素，你要知道找三个插入要比找两个插入花费更多的时间。↩
46. 只关注这个循环的位置，不关注其色向。一个色向错误的循环需要同时有其他色向错误的块（或者循环）。↩
47. 除了使用上述方法来解决这个情形，还可以通过一个棱块二交换来实现角块二交换：记住H-Perm $M_2 U M_2 U_2 M_2 U M_2$ 加多一步 U_2 就可以转变为一个角块二交换。↩ 还可以利用一些诸如 $(R U R' U')^3$ (triple sexy) 或者 $(R' F R F')^3$ (triple sledge) 来交换两组角块。↩ 这些另类的解法可能会在某些情况下很有效，但常规的解法基本总是最好的。↩
48. 除了第一种情况，你必须将那个色向错误的角块放在循环里。↩
49. 要注意到，在循环中块和面是有区别的。↩
50. 这个解法被改编过（而且少了一步），所以并没有原解法中所用到的预打乱技巧。这个技巧会在后文解释。↩
51. 最好的选择可能不止一个（比如说你找到两个可以消去3步的循环）；如果你时间不充裕，可以随机选择一个；如果你时间很多，你可以试着都尝试一次。↩
52. 想看更有趣的读物可以看这篇发布在speedsolving.com论坛FMC thread上的帖子。在这篇特别的帖子上中，严谨地证明了如何找到解决五循环的三循环组合的规则。这里还有另外一篇文章，上面的解释跟我在这一章给出的一样（也是我学到这门技术的地方）；然后再提供一个这个方法的实例。↩ 数学爱好者们可能会比较熟悉这个主题：这或多或少跟微扰理论有关。↩
53. 有些是10步+AUF。↩
54. 如果剩下不是八个块，或者不是两个四循环，可以使用"反转NISS"，后文会解释。↩
55. 大部分来源于这里，根据我的个人观点稍微做了一点调整。↩

在现有解法中插入步骤

如果你执着于一个好开头，可以试试这个：一步步回退你的解法，找到某个层没有出现所有已完成块的时机，然后你就可以对这个层进行转动（这不会破坏你后面已完成的部分），有3种选择（比如 U 、 U' 和 U^2 ），这样就通过增加一步获得3种不同的开头。这一步的代价可以获得更好的后续解法！

碰碰运气！

很明显，运气不是什么需要学习的技巧，但记住在最少步里你很需要它：长度相同的跳顶层“简单”解法并不比一个运气不好而很复杂的解法差。这是你要尽可能地多尝试的原因之一：试100次跳步的可能性会比10次或者20次高得多。

第一个例子：最后一组棱角对入槽

完成F2L-1之后，你可以通过插入最后一组棱角对完成前两层。这通常不是一个好的后续解法，除非你运气很好。你可以尝试你所能想到的各种棱角对的入槽方式⁵⁶来增加你碰运气的几率。

第二个例子：如何使用公式

首先，你需要能够识别公式的对称性： $F R U R' U' F'$ 这个OLL是关于S层对称的，所以你可以用 $B' R' U' R U B$ 解决同一个case⁵⁷。这么做可以给你两倍的跳步机会（case好的情况下）。

有个比较突出的例子，就是 $R U^2 R' U' R U R' U' R U' R'$ 这个OLL，这个公式和它的镜像公式 $L' U^2 L U L' U' L U L' U L$ 可以在4个方向解决这个OLL，而它们的逆公式又可以从另外4个方向解决。⁵⁸

第二，你不必看到什么case就用什么公式。你知道 $F R U R' U' F'$ 是一个OLL，但你也可以无视角块使用它。如果你做完F2L发现有两个棱色向不对，你就可以从4个方向试试这个公式，也许就跳步了或者变成一个容易的case（比如Sune，即OLL27）。

⁵⁶. 对于一个组好的棱角对，有 $R U R'$ 、 $R U^2 R'$ 和 $R' F R F'$ 三种入槽方式。↩

⁵⁷. 这个case的角块置换也是对称的。↩

⁵⁸. 这个case甚至不影响角块的排列顺序。↩

4. 高级技巧

上一章我们已经看了一些找出好解法必要的基本技巧。这章将展示更多的高级技巧。它们不是必要的，，但能帮你消除一些限制及给你更多尝试的可能性。

逆打乱

如果你找不到好的开头，你可以试试逆打乱。如果你用逆打乱找到了一个解法⁵⁹，你只需要把它反过来，它就变成了原打乱的解法。听起来很复杂，实际上非常简单。

举个例子，这是2012年Tim Reynolds创造的NAR：

打乱：D2 L2 B R2 U2 F' L2 U2 B2 L2 F' D L2 B U L' U2 L' F' R'
逆打乱：R F L U2 L U' B' L2 D' F L2 B2 U2 L2 F U2 R2 B' L2 D2

使用逆打乱求解：

R' U F' L2

F2 D' B' * D2 B

R2 F R2 F' R2

F D' F' D

在*处插入 B' U2 B D B' U2 B D'，抵消2步。

找出的解法：

R' U F' L2 F2 D' B2 U2 B D B' U2 B D B R2 F R2 F' R2 F D' F' D

但这是逆打乱的解法，原打乱解法为：

D' F D F' R2 F R2 F' R2 B' D' B' U2 B D' B' U2 B' D F2 L2 F U' R

通常会错误地认为原打乱和逆打乱毫无关联。它们其实非常相似。例如，如果你使用ZZ，你会注意到两个打乱在各种朝向时的“错”棱数量是一样多的，但在不同位置。你也会注意到两个打乱有相同的块，只不过颜色和位置不同。这条定理表述为：如果原打乱中的块X出现在块Y的位置，那么逆打乱中的块Y将出现在块X的位置。此外，已还原及原位翻转⁶⁰的部分在逆打乱中会被保留下来。“已完成”块留在原位，“移动”块⁶¹颜色会改变并出现在别的位置。

除了这个有用的技巧，如果你在开头卡住了或者想要更多尝试的可能性，下一段将有更多的介绍。

⁵⁹. 我重复一遍：在逆打乱中，举个例子，FRU' 是 UR'F'，不是 F'R'U 也不是 U'R F！↩

⁶⁰. 角块会向另一个方向翻转。↩

⁶¹. “已完成”块指像2x2x2、2x2x3这样已归位的块，“移动”块指像3x2x1、2x2x1和棱角对这样未归位的块。↩

伪块，预打乱和NISS

为了让“伪块”这个概念清晰一点，我们从下面这个例子开始：

打乱： F' L2 F2 U2 R2 B R2 F' R2 D2 U2 L' U' B' U R U L2 F2 L'

R2 F 构成一个2x2x1的块。如果能够2至3步扩展成2x2x2将会很不错，可惜需要4步（L' U B' D）太多了。不过试一下 L2 D'，你得到的不是一个2x2x2，而是一个2x2x2的伪块。我们可以想象成D层做了一个 D2，可以在最后再解决它。比如我们可以继续以CFOP的方式还原：

```
B' U2 R' U2 R2 U R
U2 F' U F U' F' U' F
L U2 L'
B L B' U' B U L U' B'
F2 D' L2 D F2 R2 D B2 D' R2
D2
```

在这个case中 D2 可以在OLL之前、OLL与PLL之间完成，但必须是F2L之后才可以。

不过，这个“伪造⁶²”的部分使其变得难以还原（不是高手将很难做到，比如辨认F2L）。有些人建议尝试用伪块还原，不过这里有个小技巧使其变得比较容易：你只需要把最后要做的步骤（这个case里是 D2）提到打乱前面⁶³即可。试一下！

打乱：(D2) F' L2 F2 U2 R2 B R2 F' R2 D2 U2 L' U' B' U R U L2 F2 L'

解法：

```
R2 F L2 D'
B' U2 R' U2 R2 U R
U2 F' U F U' F' U' F
L U2 L'
B L B' U' B U L U' B'
F2 D' L2 D F2 R2 D B2 D' R2
```

找到这么一个解法后，记得把预打乱加到解法的末尾，这才是原打乱的解法。

你可以使用不止一个预打乱。比如我就找到这么一个解法。

需要超过一步预打乱的伪块可能会有一些难理解。在上面的打乱中，以相同的方式构筑2x2x1（R2 F），就算是经常玩的玩家要观察出做预打乱 D F' 可以构筑出2x2x2也是比较困难的：

打乱：(D F') F' L2 F2 U2 R2 B R2 F' R2 D2 U2 L' U' B' U R U L2 F2 L'`

2x2x2: R2 F

如果你知道**NISS**（Normal-Inverse Scramble Switch）的话，这些预打乱其实也不难找到。我们需要后文的一些相关理论来理解这一技巧⁶⁴。

打乱与解法连起来可以看成是一个简单的步骤序列循环，它不会使魔方的状态有什么改变。举个实例：

打乱：A B C D

解法：p q r s

序列 A B C D p q r s 会使魔方回到复原态。同时，我们可以在这个序列里添加一些镜像转动：

s (A B C D p q r s) s' = s A B C D p q r

r s (A B C D p q r s) s' r' = r s A B C D

q r s (A B C D p q r s) s' r' q' = q r s A B C D

p q r s (A B C D p q r s) s' r' q' p' = p q r s A B C D

D p q r s (A B C D p q r s) s' r' q' p' D' = D p q r s A B C

...

这对魔方的状态没有什么影响。⁶⁵

我们在第一个实例中用了预打乱 D2，这个循环序列可以写成：

(打乱) R2 F L2 D' (其它步骤) D2

在前面预先加上的 D2 只是加入镜像转动的一个变形：

D2 (打乱) R2 F L2 D' (其它步骤)

换句话说，我们可以把“R2 F L2 D' (其它步骤) D2”看作是“ (打乱) ”的解法，也可以看成“R2 F L2 D' (其它步骤)”是“D2 (打乱)”的解法。⁶⁶

这大概足以理解预打乱的工作原理了。知道了这一点，我们就可以认为解法是原打乱的打乱以及和逆打乱的关系。怎么做？思考一下同样的打乱和同样的开头 R2 F。从这里开始我们解法看起来会像 R2 F (w)，这里 (w) 一个步骤序列。我们的循环序列是：

(打乱) R2 F (W)

正如前面我们所说的，它的逆序也是一个“神奇”的循环序列（相当于用逆打乱找到一个解法）：

(W)' F' R2 (逆打乱)

我们可以认为“(W)' F' R2”是“(逆打乱)”的解法，但同时，像“(W)'"也是“F' R2 (逆打乱)”的解法。因此，你可以把在原打乱找到的步骤逆过来当成逆打乱的预打乱。用加了预打乱的逆打乱找到一个解法（我们把它称为(K)，对应前面的(W)'）后，就大功告成了：最终解法就是 R2 F (K)'。

你可以重复这个过程：假设你在加了预打乱的逆打乱中找到步骤 F D'（做了一个2x2x2），但没有好的后续解法，这时我们可以回到原打乱，把 D F' 作为预打乱。实际上，循环序列是这个样子的：

F' R2 (逆打乱) F D' (之前找到的步骤)

反过来变成另一个神奇的循环序列：

(之前找到的步骤的逆序) D F' (打乱) R2 F

所以我们可以把 D F' 作为原打乱的预打乱，然后以 R2 F 开头。

这里给出一个[实例](#)，可能会比较清晰易懂：

打乱：(fmc.mustcube.net #265) F2 B' R' B D B F' R B F L' D2 U2 B2 D U' B' F D2 L' F2 B' R2 B2 R L' F2 U' F' L
解法（作者Guus Razoux Schultz）：D' B' U B2 D' L F2 L' D R U2 R U2 B' R' U2 R U' R B' R' F2 U' L B'（25步）

解释：

逆打乱的一个好开头：B L' U F2

回到原打乱：[预打乱F2 U' L B']不错的后续解法：D' B' U B2 R2

继续来到逆打乱：[预打乱R2 B2 U' B D]：B L' U F2

逆打乱中简单的后续：

F2L: R B R' U R' U2 R B

LL: U2 R' U2 R' D' L F2 L' D R2

修正预打乱：R2 B2 U' B D 抵消2步（25步）

为了使写这样的解法更简洁明了，我提出以下记录转动的方案⁶⁷：把在逆打乱中找到的解法用括号括起来。这样Guus的解法就变成：

好开头：(B L' U F2)

不错的后续解法：D' B' U B2 R2

F2L：(R B R' U R' U2 R B)

LL：(U2 R' U2 R' D' L F2 L' D R2)

转动表示简单易懂，避免了“加了预打乱的原打乱/逆打乱……”这类重复。

⁶². 我希望我把它的意思解释清楚了。↩

⁶³. 这就是为什么把它称为“预打乱”。↩

⁶⁴. 来自岡山友昭（Tomoaki Okayama）的[这个回复](#)。↩

⁶⁵. 逆序列也不会改变魔方的状态。↩

⁶⁶. 你也可以把打乱看成是解法的解法。↩

⁶⁷. 这种表示方法并不是很流行，因此你可以在旁边给一些简单的提示，比如（括号内的步骤是在逆打乱下进行的）。↩

反转NISS

这不是一个应用很广泛的实用技巧，只是偶尔使用到。它可以被认为是“共轭与还原”和“在现有解法中插入步骤”的改进版。假设你找到了一个好的构造解决了除少部分块（4到8块）外的其它部分，你大概会想插入公式来解决它。不过，如果这些块不具有至少一种共同的颜色（不在同一层），将会很难识别到底该用什么公式。这里有个小技巧：你可以在做解法的时候跟踪未完成的块，如果发现在某个点它们出现在同一层，你就可以在这个点后下一个“断点”，把后面的步骤都作为预打乱（这就是我称之为“反转NISS”的原因）。然后，你就只剩下“最后一层”没有完成。有需要的话可以做一些setup。我们来看看这个实例：⁶⁸

打乱：L2 D2 F2 U' L2 D L2 D2 B2 U' F' U' R' D B2 L D B' F' R'

解法：F2 U R' U' F D2 F R F D F2 R F R2 D R D2 R' F2 U2 R D R' D2 B2 R2（26步）

解释：

除了5块外的其它部分：F2 U R' U' F D2 * F' U2 R D R' D2 B2 R2（把R2作为预打乱就很容易看出来）在* 处插入F R可以把未完成块归到一层。我们来看看这个构造：

F2 U R' U' F D2 F R + R' F' F' U2 R D R' D2 B2 R2

然后在 + 处“下断点”：把R' F' F' U2 R D R' D2 B2 R2当作原打乱的预打乱，现在变成：

预打乱：R' F2 U2 R D R' D2 B2 R2

F2L：F2 U R' U' F D2 F R

LL：F D F2 R F R2 D R D2

⁶⁸ 事实上，在这个解法中，公式很容易通过标记箭头和叉识别，所以我没有使用这个技巧，但这是个很好的例子。↩

实用公式

你可以看到，上一个实例中我用到了常用的OLL：模转动， $R\ U\ R^2\ F\ R\ F^2\ U\ F\ (U^2)$ 。有的时候这很实用，因为它是最短的只改变色向而不改变位置的公式。

一般来说，一些最短的，**9步**或**10步**的最后一层的公式是很实用的。你可以在[这里](#)找到一个完整的列表（模转动，对称的和逆序的）。⁶⁹

如果你决定不遵守“不要只顾着还原F2L而不去理会最后一层”这一规则的话（通常不是个好的尝试！），可以指望最后一层的公式很短：实际上你会的越多就越好！⁷⁰

这里还有两个为什么学一些公式（至少是**6到9步**的）是值得的原因：第一，如果你不知道最后一层的确切公式或者F2L还没完成，这些公式其中的某一个可能会给你一个好的构造（比如一个三角循环）。

另外，学习这些公式可以教你怎样构筑块或者完成F2L。让我们来看一个不错的T perm：

```
R2 Uw R2 Uw' R2 y L2 Uw' L2 Uw L2 (U')
```

它就是一条实用的F2L公式叠加了两次。

除了学习公式，你还可以用现有的公式衍生出新的公式。

⁶⁹. 有些公式有自己的名字和注释，就个人使用来看，你完全可以放心地忽略掉这些。我会在以后发布一个比较好的注释和排序列表。 [↩](#)

⁷⁰. 使用最后一层的公式时，可以在公式之前或之后做AUF（“Adjust Upper Face”，顶面调整），前面有预打乱的话说不定就可以消步了。 [↩](#)

棱角对分析

这是一个毫不起眼的技巧，全靠直觉，也没有什么能证明在实际应用中能给你带来优势。它到底是什么？“分析棱角对”意思是对打乱进行一些分析，比如：

- 已完成块，这个不用多说。
- 可以一步完成的棱角对。你可以使用“暴力”法很轻松地将它们找出来。此外，它对如何构筑伪棱角对（可以在逆打乱中一步完成的棱角对）也有帮助。能够在原打乱或者逆打乱中一步构筑棱角对可以作为我们的第一步，也可以考虑把这一步作为逆序的预打乱。
- 错误的棱角对：有些棱角对因为有一块色向错误而很难构筑。这种棱角对你就尽可能地去破坏吧。

这种技巧，特别是错误的棱角对，没有多少文档提及到。Guus Razoux Schultz对[这个打乱](#)做了不错的分析。

5. 如何练习

许多人说，想要进步的话，你需要“练习，练习，再练习”。这是事实，不过，你也需要知道怎样练习：这里有一些如何练习最少步的建议。

不要限制时间和模拟比赛

经常尝试模拟比赛并迫使自己在一个小时内完成解法并不是一件好事。于此相反，我建议不要限制时间且多次尝试同一个打乱，直到你对结果满意为止。

并不是说一个小时内完成解法不好。它可以明确你的水平，帮助你找到一个好的时间管理策略⁷¹。对于这种训练，我建议参加[David Adam的线上周赛](#)。

这里有个折中的办法，就是先像比赛一样尝试一个小时，之后再继续尝试直到有满意的解。

⁷¹. 第7章我会给一些建议。↩

和高手比较（并学习他们的解法）

当你想练习的时候，我建议去尝试已有最少步高手给出解法的打乱，这样你可以比较你的解法和他们的区别，找到你错过的好开头或其它东西。

此外，学习高手们的解法可以潜移默化地训练块构筑和其它解法。你可以在fmcsolves.cubing.net上面找到很多。

困难的打乱

看看自己可以怎样应付“糟糕的case”。我建议试试一些高手们认为很难的打乱。你可以在[这里](#)找到一些困难的打乱。

针对性练习

如果你在寻找好的开头方面有困难，尝试一下练习：拿一个打乱，找2x2x2、2x2x3或者其它东西，直到你自己满意，然后换一个打乱继续。你可以把这个方法推广到F2L-1或其它步骤。

快速打乱

尽管这可有可无，不过当使用如NISS等技巧，或者需要多次重新打乱进行尝试时，你应该至少做到打乱“不会太慢”⁷²，并且最重要的是准确（不出错）。

⁷². 20步打乱在10秒内就差不多了。↩

学习！

最后再强调一点，可以学习这个教程，或者从一些其他的资源里学习公式、技巧，了解一些技术和知识。我已经把speedsolving.com的[The FMC Thread](#)从头到尾看了两遍。

学习公式：有许多不同的公式集，比如[LLEF](#)（Last Layer Edges First）或者[Summer Variation](#)。记住，你不仅仅要记住它们，还要试着理解它们如何工作。

6. 一些其他方法

“标准”方法是通过块构筑来构造并用插入来解决剩下的少部分块，不过还是有其它一些值得一提的方法。

用EO开头

你应该记住这种可能：从调整全部棱块色向开头（**EO**，**Edge Orientation**），比如ZZ。这里你有两种思路来继续后面的解法。

EO + 块构筑

调整完所有棱块色向之后，最常用的方法是块构筑。这么做的好处是没有了“坏棱”，缺点⁷³是不能做任何会破坏EO的步骤，这会有一些（相对较小的）限制。

Grzegorz Łuczyna通常会以EO开头。以下是他在European Champion 2010上给出的解法：

```
打乱：L D2 B' D2 B R' B' U B L B L2 B2 U2 F2 U R2 D' B2 D' B2
解法：
EO: x y2 L2 D F'
EOline: R L2 D #
2x2x3: R' U2 B2 R2 B2
剩3角：L2 U' R' U L U' L' R U2 L' U'
3角插入：[D2, R' U' R] at # (4 moves cancel)
```

“Domino”还原（和HTA）

EO可以被认为是一种模转动，即通过< R, L, U, D, F2, B2 >把魔方降解到一个子群，再通过< U, D, R2, L2, F2, B2 >降解到下一级子群。想要这么做的话，你必须：

- 把E层棱块归位到E层
- 还原角块色向

这种还原方法也被称为“Domino”，因为它把三阶魔方变成了3x3x2魔方（Domino魔方）来进行还原。此外，这也是Human Thistlethwaite Algorithm（HTA），一种Thistlethwaite Algorithm的修改版本的前两个步骤。如果你对这种最少步方法感兴趣，可以看看这个[教程](#)。

这里有Per Kristen Fredlund的一个实例：

```
打乱：D U' R' F B2 R B2 R' U2 R B2 R' U2 R B2 R' U2 R B2 R' U2 B2 F2 U D F D' B2 D F D'
      B2 D F D' B2 D F D' B2 F' D B' F R'
解法：
R' B U' D F //EO (5/5)
L' F2 L //Domino还原(3/8)
D2 L2 F2 D F2 D L2 U' R2 D2 R2 //完成 (11/19)
```

73. 其实没有人强迫你这么做，不过做了EO之后破坏它看起来并不是一件明智的事。如果你愿意，以部分EO开头也不是不可以。 ↩

角先

“角先”（Corners First，有时简称CF）并不单指一种方法，而是先还原角块再还原棱块的这一类方法的合称。桥式在一定程度上可以算是角先法。

一部分人便是从角先法开始学习还原魔方的。⁷⁴分开还原角块和棱块使解决思路更加简单和直观。此外，先还原角块可以更加自由地还原棱块，同时只转动内层也不会影响到角块。

但是角先法用在最少步中也有缺点：内层转动一步是记为两步的！尽管如此，还是有至少两位最少步高手在使用：只参加过两次官方比赛的Attila Horváth⁷⁵和西班牙NR保持者Javier Cabezuelo Sánchez⁷⁶。他们都认为角先法很适合解最少步，但不太适合一小时限时赛。⁷⁷

Attila Horváth通常使用类似Guimond（色先）的方法还原角块，这一步不用理会中心。他有时会在这步使用预打乱或者NISS。接下来他会在角块解法里将部分单层转动改成双层转动或者插入中层转动使至少2到3个棱块归位。之后解决剩下的棱块，没有固定的顺序。他有时会到最后才用插入⁷⁸解决中心，通常可以消去很多步。[实例](#)

Javier Cabezuelo Sánchez用另一种思路还原角块：先还原第一层的角块再解决其它的。然后他会在角块解法中插入步骤（或者公式）来还原棱块。他不使用诸如逆打乱、预打乱或NISS之类的技巧。和Attila不同，他在角块还原的时候会一并还原中心。⁷⁹

Attila和Javier两人都只使用他们的角先法而没有学习其它方法，这违背了“不要自我限制”这一规则，不过，他们仍然取得了不错的成绩。

⁷⁴. 例如Valery Morozov就做了一个他的方法的[教程](#)。↩

⁷⁵. WCA主页：[2012HORV01](#)；speedsolving.com简介：[Attila](#)。↩

⁷⁶. WCA主页：[2007SANC01](#)。↩

⁷⁷. 要注意到Javier的很多解法都是DNF。↩

⁷⁸. 像 $M E M' E'$ 或者 $M E^2 M' E^2$ 这类公式。↩

⁷⁹. 来源：[这个回复](#)。↩

7. 比赛

如何写解法

无论比赛还是练习，解法里面不要出现整体转动。这么做有以下好处：

- 使用整体转动更容易出错。
- 整体转动会影响消步。⁸⁰
- 还原时，整体转动魔方会使你需要分散精力去注意转动了哪些层。

如何写解法而不出现整体转动呢？比如写B面的PLL就会很别扭。这里有一个比较简单的办法：在标准配色下保持标准朝向，只要记住白色中心是U，绿色中心是F，黄色中心是D，以此类推⁸¹。当你转动的时候，比如白色中心所在的层，不用管它朝向那里，写U就是了。

⁸⁰. 举个例子，如果你写出像 $R\ Z'\ U'$ 这样的解法的话会闹出笑话的。↩

⁸¹. 为了帮助记忆（虽然不是很难），记住蓝（Blue）和红（Red）的首字母和它们所在层是一样的。↩

备用解法

在限时比赛中写一个“备用解法”是一个好习惯。它通常不是一个多好的解法，但总比DNF了强。如果在最后时刻才匆匆找一个，有可能会出错甚至就找不到解法了。举个例子，你的平均大概是35步，你20分钟时找到一个40步的解法并记下来⁸²，剩下的40分钟你可以更冷静地去思考。这里提供一些合理的找备用解法的途径：

- 促使自己在一定时间内（比如35分钟）找到一个解法，不管多糟糕都把它记下来。我个人不会这么做，但如果你经常发现自己到最后都写不出什么的话这或许会有些用处。
- 如果你随便做的时候找到一个解法（比如你找到一个好的开头，在还原准备重新打乱的时候跳P了），把它记下来。
- 我是这么做的：我不会真的去找一个备用解法，但会找一些备用构造。举个例子，我的目标通常是sub30，在一个case中我找到了一个26步剩3角的构造，虽然不是很好但也把它记下来，到只剩下10分钟时如果我还是找不到更好的构造我就用它来找插入。⁸³

什么是好的备用解法？任何解法！任何解法都比DNF强，特别是现在的最少步比赛（官方比赛）取3次平均，有一个DNF的话你的平均就DNF了。

⁸². 你甚至可以写在比赛答题纸上。如果最后你想改解法，划掉重写即可。↩

⁸³. 我插入一个三角换一般需要5分钟。↩

时间安排

“怎样安排你的时间”是一个复杂的话题，我不能说我的建议在任何情况下都是对的。个人经验是最好的老师。

不要太执着！

任何人都会碰到这样的情况：在比赛中确定了一个开头之后找不到比较好的后续。快速判断一个开头是否会有好的后续解法的能力，就和读懂别人想法的能力一样有用。我有一个或许微不足道的建议：不要太执着。如果你试了所有你会的方法和技巧后还找不到解法，别盯着魔方发呆指望它自己解决问题，赶快回去试试别的。

尝试所有可能

如果你是一个计算机科学家、数学家什么的，我可以告诉你，尝试不同的可能解法其实就是一个树状搜索：你可以选择DFS（深度优先，尝试完成一个解法后再做另一个）、BFS（广度优先，同时进行几个解法）方法或同时进行。要注意，单个部分解法可以衍生出一堆不错的分支也可能什么都没有。基于这个原因，我不会使用固定的方法。另外，知道什么时候舍弃掉无用的分支（指一些没有发展前途的部分解法）也同样重要。

我希望非专业人士也能看得懂我写的是什麼。

8. 其它资源

作为补充，这里列出了我获取所有这些信息的其它有用的链接：

- [David Adams的网站](#)，这里有举行周赛，你可以和不同水平的玩家进行较量，可以很好地学习到同一打乱的不同解法。
- [fmc.mustcube.net](#)，作者Per Kristen Fredlun，现在下线了，以前用来举办周赛。
- [Insertion Finder](#)，对找插入很有帮助。
- [JARCS](#)，一个用来寻找（以及学习）解法的常用的子步骤，可离线使用。
- [fmc solves.cubing.net](#)，最少步解法的博客/数据库，官方的和非官方的都有。
- [menas.com.br](#)，巴西玩家制作的官方解法搜集网站，每一个都有解法讲解视频（葡萄牙语）。
- [Speedsolving.com](#)，对玩魔方有很好的参考价值，特别是：
 - [The FMC thread](#)，专注于最少步还原。
 - [Fewest Moves: Tips and Techniques](#)，另一个关于最少步的板块。
 - [Brian Yu的循环教程](#)。
- [Ryan Heise的网站](#)，解释了他的方法和一些“[基本技巧](#)”。
- [Lars Petrus的网站](#)，包括一些块构筑的实例。
- Daniel Sheppard的一个[视频](#)，提出了一些关于如何进步的建议（译注：需翻墙）。
- Ranzha的5个视频教程（译注：需翻墙）：[\[1\]](#) [\[2\]](#) [\[3\]](#) [\[4\]](#) [\[5\]](#)
- Pranav Maneriker的一个[Prezi演示文档](#)。