

Instructions for Solving Rubik Family Cubes of Any Size

Ken Fraser¹

Date of original document: 25 September 2007

Date of this revision: 12 February 2017

Summary

The purpose of this document is to provide instructions on how to solve Rubik's family cubes of any size. Only standard cubes with unmarked centres are covered in this document. The instructions are provided as printable character sequences (most of which are often referred to as algorithms) that can be applied to any Rubik family cube implementation.

The instruction set used is that which applies for the Unravel software application. That application utilizes keyboard data entry to control cube layer rotations and has to conform to a simple text format (no subscripts or superscripts). The cube solving methodology described herein can be used for any hardware or software implementation although translation to other instruction sets may be desirable for some people.

For Unravel program users, particularly beginners, the on-line help provided when that program is run may not give sufficient advice on all the steps involved in unscrambling cubes. This document is meant to complement the on-line help in such cases.

¹ Ken Fraser retired in 2002 as Principal Research Scientist and head of Helicopter Life Assessment at the Aeronautical and Maritime Research Laboratory (as it was known at the time), Defence Science and Technology Organisation, Department of Defence, Australia. This publication is the result of a leisure activity and has no relation to work at the Laboratory.

Contents

	Page
Definitions	iii
1. Introduction	1
2. Keyboard Commands	2
3. Cube Size Implications	4
4. Proportion of Cube to be Set Without Macros	4
5. Placement of Bottom Layer Centre and Edge Cubies	5
6. Placement of Remaining Lower Centre Cubies	5
7. Placement of Upper Centre Cubies	8
8. Placement of Bottom Layer Corner Cubies	11
9. Alignment of Remaining Eight Edge Cubie Sets	11
10. Final Layer Actions	16
11. Centre Cubie Block Transfer Macro	19
References	21
Appendix A: Algorithms as Defined in Macros for Use with Size 16 Cube	22
Appendix B: Using Macros for Placement of All Centre Cubies	30
Appendix C: Laws of the Size n Cube	32
Appendix D: Final Layer Problems for Cubes of Size 4 and Above	34

Definitions

Cube size	The standard Rubik's cube is often referred to as a 3x3x3 cube. That cube will be referred to as a size 3 cube and in general an $n \times n \times n$ cube will be referred to as a size n cube.
Rubik cube family	Cubes that have similar rotational properties to the standard Rubik's size 3 cube and obey generalised rules for a size n cube are considered to be members of the Rubik cube family. Cubes of size 2 and above that meet this condition are available.
Cubie	Individual cube elements will be referred to as cubies (others sometimes refer to them as "cubelets"). There are three types of cubies: corner cubies (three coloured surfaces), edge cubies (two coloured surfaces) and centre cubies (one coloured surface). The absolute centre cubies for odd size cubes sit on the central axes of each face and their relative positions never change.
Cubicle	A cubicle is the compartment in which a cubie resides. For a permutation, cubicles are considered to occupy fixed positions in the space occupied by the cube object but their contents (cubies) may shift position.
Facelet	A facelet is a visible coloured surface of a cubie (corner cubies have three facelets, edge cubies have two and centre cubies have one).
Cube state	A particular arrangement of the cubies will be referred to as a cube state. What looks the same is considered to be the same (unless specific mention to the contrary is made). Each state has equal probability of being produced after a genuine random scrambling sequence. A rotation of the whole cube does not change the state considered herein. In other texts the various states are often referred to as permutations or arrangements.
Cube layer	A cube layer is a one cubie width slice of the cube perpendicular to its axis of rotation. Outer layers contain more cubies than inner layers. For a cube of size n there will be n layers along any given axis.
Cube face	The meaning of a cube face depends on the context in which it is used. It usually means one of the six three-dimensional outer layers but can also refer to just the outside layer's surface which is perpendicular to its axis of rotation. The faces are usually designated as up (U), down (D), front (F), back (B), left (L) and right (R).
Set state	The set (or solved) state of the standard cube is one for which a uniform colour appears on each of the six faces.
Scrambled state	The scrambled state is the starting point for unscrambling the cube. It arises when a cube in the set state is subject to a large number of randomly chosen layer rotations.
"Fixed-in-space" axes of rotation	There are three mutually perpendicular axes ^[1] of rotation for the cube. One set of axes defined in terms of D, U, B, U, L and R terms can be considered to have a fixed orientation in space. Think of these axes as belonging to a cube-shaped container in which the cube object can be positioned in any of 24 orientations. One axis can be drawn through the centres of the D and U faces (the DU axis). The others are the BF and LR axes.
"Cube object" axes of rotation	Another set of axes can be defined ² for the cube object itself. These axes relate to the face colours, the most common being red, green, orange, yellow, white and blue. The axes are usually white-blue, red-orange and yellow-green. The origin for the axes is the centre of the cube object.

² For odd size cubes these axes are always fixed relative to the internal frame of the cube object. For even size cubes these axes remain fixed relative to the internal frame of the cube object after initial selections.

Layer rotation	The only way that cube state can be changed is by rotations of cube layers about their axes of rotation. All changes of state involve rotation steps that can be considered as a sequence of single layer quarter turns.
Orbit	<p>For a basic quarter turn of a cube layer for cubes of all sizes, sets-of-four cubies move in separate four-cubicle trajectories. When all the possible trajectories for a given cubie type are considered for the whole cube we will refer to all the possible movement positions as being in a given orbit. We consider that the size 3 cube has two orbits, one in which the eight corner cubies are constrained to move and one in which the 12 edge cubies are constrained to move. Transfer of cubies between these orbits is impossible.</p> <p>For cubes of size 4 and above we will also define an edge cubie orbit as comprising 12 cubies but will use the term <i>complementary orbit</i> to describe a pair of orbits between which edge cubies can move. A pair of complementary edge cubie orbits contains a total of 24 cubies. Cubes of size 4 and above include centre cubie orbits that contain 24 cubies. Transfer of cubies between one such orbit and another is not possible (applies to cubes of size 5 and above).</p>
Move	A move is a quarter turn rotation of a layer or a short sequence of such quarter turns.
Move notation	A clockwise quarter turn of an outer layer is usually expressed as U, D, F, B, L or R. In other respects the notation used varies among authors.
Algorithm	An algorithm defines a sequence of layer rotations to transform a given state to another (usually less scrambled) state. Usually an algorithm is expressed as a printable character sequence according to some move notation. An algorithm can be considered to be a “smart” move. All algorithms are moves but few moves are considered to be algorithms.
Permutation	<p>A permutation^[2] of the cube as used herein means the act of permuting (i.e. rearranging) the positions of cubies. A permutation is an all-inclusive term which includes a sequence of quarter turns of any length. Under this definition a cube move or rotation sequence is a permutation. Even the solving of the cube from a scrambled state represents a permutation. The term “permutation” is used extensively by mathematicians who use Group Theory^{[3][4][5]} to quantify the process involved in a rearrangement of cubies.</p> <p>The term "permutation" is also often used to mean the state of the cube that results after it is rearranged but that meaning will not be used herein. In such cases the term “cube state” will be used. That allows the term “permutation” to be used when the permutation results in no change of state – an area of special interest for Rubik’s family cube permutations.</p>
Parity	A cube permutation can be represented by a number of swaps of two cubies. If that number is even the permutation has even parity ^[6] , and if the number is odd the permutation has odd parity.

1. Introduction

For the Unravel^[7] application developed by the author, cube size is selectable in the range 2 to 16 for the C++ version, whereas the Java version has the same lower limit but has a settable upper limit ranging upwards from 16 but never exceeding 99. The upper cube size limit for the Java version is dependent on screen resolution and the user's visual acuity. For many users a size 16 upper limit is likely to be adequate.

On-line help provides the main source of help for running the Unravel application. This document is an extension of that available on-line. Central to the cube solution software is the adoption of keyboard text entry as the means by which the user manipulates the state of the cube. To facilitate the process of solving the cube, macros³ have been provided for cubes of all available sizes. A macro allows shorthand text entry for implementing a sequence of commands. If, for example, the character x is never used for basic input we could create a macro $x = abcd$. Whenever the user types x , the program would substitute $abcd$ for that. In Rubik's cube parlance, command sequences that, in the main, perform non-trivial transformations are often referred to as algorithms. Most people either memorize algorithms or look them up in a printed reference. Macros can be thought of as algorithm shortcuts. While macros cannot be used for solving hardware cubes or software cubes other than Unravel, the macro definitions can be used for solving those cubes in the standard way.

On-line General help is focused mainly on what the program does and how the user interacts with it. On-line Macro help is focused mainly on the definition of macros and what they do. Many of the earlier steps in unscrambling a cube are fairly simple and a high proportion of users may not use the full set of macros provided for the Unravel program. It is likely that users of the Unravel program would apply macros for unscrambling 50% or more of the cube.

The on-line help does not attempt to provide guidance on how to unscramble the portion of the cube for which macros are not provided and on how to handle some tricky unscrambling steps that may be required in conjunction with the use of some macros. Special attention is given in this document in those areas. Details on the algorithms used for all macros for the Unravel application are provided herein and a sample set of algorithms (as defined for the default macros) for use with a size 16 cube is provided in Appendix A.

Macro comments herein apply to the default macro set provided for both Java and C++ versions of the Unravel software available from the author's website, but the cube size ranges provided differ, as indicated above, for these versions. Users can define their own macros, but in such cases this document may have reduced applicability. Macros make solving the cube simpler and faster for most people but their use is optional. There are many different algorithms that can be invoked for solving cubes and those used for the macros defined for the Unravel application represent just one, by no means unique, approach.

Solving the cube, with the help of the macro set provided in the on-line macro help, means some restrictions may apply to the order in which the cubies are placed.

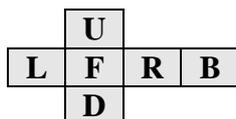
The Java version of the Unravel program provides a marked centre cubies' option for the cubes in the 3 to 32 (typically) size range. Algorithms (or macros) defined in this document are also applicable to the marked option but additional algorithms are needed for solving those cubes. Similar to the standard cube with unmarked centre cubies, the Unravel program on-line help

³ The term "macro" as used in computer science is a rule or pattern that specifies how a certain input sequence should be mapped into an output sequence. Macros are normally used to map a short string to a longer string (sequence of instructions). Macros simplify things in Unravel by providing short-cuts for long sets (or frequently used short sets) of rotational instructions to produce a desired change of cube state.

provides the main source of help for solving cubes with marked centre cubies. Additional help for Unravel program users, or users of any other implementation that involves marked centre cubies, is provided in another document^[8] written by the author.

2. Keyboard Commands

Basic (outer layer) rotation symbols are D (down), L (left), F (front), R (right), B (back) and U (up). These symbols correspond to those most commonly used by authors of cube solving documents. The Unravel application displays the six faces of the cube in a two-dimensional manner as illustrated below.



Other alphabetical characters (except M and W) can and, in most cases, are used for macro definitions. Further macros, which will be referred to as numerical macros, are defined by placing a number within angle brackets (or without brackets if the character M precedes the number). These macros conform to the arrangement $\langle (Row)(Column) \rangle$ where *Row* and *Column* conform to a coordinate structure that facilitates their use particularly for centre cubie placement. For convenience, a row digit will be designated as *R* and a column digit as *C*. For cubes of size up to 16 (for both C++ and Java versions), a total of two digits is used, whereas for cubes of size greater than 16 (only applicable to Java version) a total of four digits is used. That results in numerical macros of the form $\langle RC \rangle$ (two-digit) and $\langle RRCC \rangle$ (four digit) respectively. For the two-digit type, *RC* is a hexadecimal number⁴ in the range 00 to FF (although the row and column upper limits are dependent on cube size for the Java version). For the four-digit type, *RRCC* is a decimal number.

The numerical macros are useful when applied to positioning of cubies that occupy a coordinate grid (centre cubies for example). A *macro command* refers to a single character (e.g. P) or multiple characters for two-digit macros (e.g. $\langle 27 \rangle$ or M27) and four-digit macros (e.g. $\langle 4527 \rangle$ or M4527) that need(s) to be entered via the keyboard to represent a macro.

When one of the six basic command symbols (i.e. D, L, F, R, B or U) is entered, a clockwise (CW) quarter turn (90 degrees) rotation of the specified outer layer occurs. When a basic command symbol is preceded by a decimal digit *n*, in range 2 to 16 for the size 16 cube for example, the base and all layers up to that number are rotated 90 degrees CW about the relevant axis of rotation. Thus *nR* would rotate *n* layers about the R face axis. If *nR* is preceded by # or 0 (zero) only the *n*th layer is rotated. Where multiple layer rotations are required, it is important that single digit multiple layer rotation numbers (in range 2 to 9) are not preceded by a leading zero as the leading zero will be interpreted as a single layer rotation identifier. For a half turn (180 degrees) rotation, the command needs to be entered twice. To produce a counter clockwise quarter turn (−90 degrees) rotation, the command needs to be preceded by the “−” (minus) sign (character code 2D hexadecimal). A minus sign in front of a macro (for example −P or −M27 or −M4527) implements a reversal of the macro sequence. Thus the sequences P−P and −M27M27, for example, will result in no change of cube state. Reserved symbol W in front of a base symbol rotates the whole cube. Thus for a size *n* cube, WR is equivalent to *nR*, and will rotate the whole cube 90 degrees CW about the R face axis.

Move notation varies among authors, especially in the way single and multiple layer rotations are presented. Hence there was no “standard” approach to follow in the Unravel software. Most

⁴ Hexadecimal numbers from 00 to 1F (0 to 31 decimal):
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F

commonly, other authors designate a CCW 90 degree rotation by a dash after the rotation symbol (e.g. F'). The use of the minus (e.g. -F) ahead of the rotation or macro symbol was considered an advantage for the Unravel software, which has an option to provide an immediate update of the cube state whenever the user enters a valid command. If a post-rotation or post-macro symbol were used, such as F', a CW 90 degree rotation would be displayed after the F was typed and then a 180 degree rotation would be observed when the ' was typed. The end result would still be correct but use of -F rather than F' would eliminate the superfluous rotation.

Optional macro symbol M in lieu of < > and single layer rotation symbol 0 (zero) in lieu of #, would normally be used for keyboard entry. In that way the cube can be completely solved without the need to use the shift key (lower case alphabetical characters are always internally converted and displayed as upper case). Symbols <, > and # improve readability when used in help text and have been adopted for the macro help text available on-line. Symbol # is used in this document. While many macro commands have the same definition irrespective of cube size, some don't. The Unravel application generates a different default set of macros for each cube size. Sample macro definitions are provided below.

Macro definition	Comment
P = FUURU-RUU-F	This is an alphabetical macro to be translated as: F: Rotate the Front outer layer by 90° CW U: Rotate the Upper outer layer by 90° CW U: Rotate the Upper outer layer by 90° CW R: Rotate the Right outer layer by 90° CW U: Rotate the Upper outer layer by 90° CW -R: Rotate the Right outer layer by 90° CCW U: Rotate the Upper outer layer by 90° CW U: Rotate the Upper outer layer by 90° CW -F: Rotate the Front outer layer by 90° CCW
<23> or M23 = #3B#2U-#3BU#3B-#2U-#3BUU	This is a numerical macro. Note the use of #3B here. This instruction means that only the 3 rd layer from the Back face is rotated 90° CW. The other layers on the Back face axis are not rotated.
Q = P-UP	This is an alphabetical macro which includes a call to another macro.

Editing of the input command sequence by Backspace or Delete keys can be used as and when desired. It is expected that most users would select the option "*Update cube display after each command*". When that option is selected, there may be a tendency to enter commands ad infinitum. However, if more than 60 characters are entered before ending the sequence by pressing the <enter> key, a "*Command sequence too long*" error will be flagged when the <enter> key is pressed. Any error in the command sequence will halt immediate update of the state of the cube and can be corrected, if desired, by repeated entry of Backspace until the cube state changes, which will occur after the last error is removed.

For both C++ and Java versions of the Unravel program there are 18 alphabetical symbols (A, C, E, G, H, I, J, K, N, O, P, Q, S, T, V, X, Y and Z) that can be used as macro names (N and O are not used in the default sets provided). For the C++ version a fixed number (256) of

numerical macro names are available for any cube in the size range provided. For the Java version, the number of numerical macro names that can be used varies with cube size. For example, macro names M00 to MFF (two-digit number is in hexadecimal form) can be used for cube size 16 for the Java version and any size for the C++ version. Spare (unused) numerical macro names are available when the default sets are used.

3. Cube Size Implications

Once all the techniques for solving cubes up to and including size 5 are fully understood, larger size cubes can be solved using more of the same steps. Macros for movement of corner cubies are the same for cubes of any size. Matched sets of edge cubies behave the same as those for the size 3 cube and all macros defined for the size 3 cube are applicable to any other cube of larger size. However, the matched edge cubie sets for even size cubes of size 4 and above, can occupy two arrangements that are not possible for cubes of odd size. These are:

- A final layer requiring an odd number of edge cubie set flips (can be corrected using default macro S).
- A final layer requiring the positions of a single pair of edge cubie sets to be swapped (can be corrected using default macro X).

For cubes of size 5 and above, simple matching of edge sets using macros P and Q can result in a final set that is not a matched set (the sense of cubies in the set varies). Macros are provided to resolve this. The required corrections noted above, which are sometimes referred to as parity or last layer (OLL⁵ and PLL⁶) corrections, are considered further in Sec. 10 and Appendix A.

4. Proportion of Cube to be Set Without Macros

Macros (or algorithms) may have limited application in the early stages of unscrambling a cube where very simple moves can be applied to achieve the desired result. It is likely that many users would unscramble about 40% of the cube without the use of macros. In such cases users would be likely to put in place the following without the use of macros.

- All bottom layer cubies except the corner cubies.
- Other centre cubies below the horizontal centre line through L, F, R and B faces (for even size cubes the number above and below this line will be the same, whereas for odd size cubes this line will bisect a layer of centre cubies for which macros are defined).

Some users may find the defined macros useful for placing all centre cubies (Appendix B).

For a cube of size n :

Number of corner cubies	=	8
Number of edge cubies	=	$12(n - 2)$
Number of centre cubies	=	$6(n - 2)^2$
Number of facelets	=	$6n^2$
Total number of cubies	=	$6(n - 1)^2 + 2$
Change in total number of cubies for unit change in cube size from n to $n + 1$	=	$12n - 6$

⁵ An OLL (orientation of last layer) requirement refers to a need to flip a complementary pair or a complete set of edge cubies in the final edge set.

⁶ A PLL (permutation of last layer) requirement refers to a need to swap the positions of two edge cubie sets in the final layer.

Table 1 illustrates how dominant centre cubies become as cube size is increased. For cubes of large size, most of the macros provided and most of the time spent solving the cube relates to the placement of centre cubies.

Table 1: Centre cubies as a proportion of total cubies

Cube size	4	8	16	32	64
Centre cubies as a proportion of total cubies (%)	42.8	73.0	87.0	93.6	96.8

Table 2 gives values for the total number of cubies and number of cubies to be set without macros, for the method described herein in Sections 6 to 10 for the Unravel program, as a function of cube size for cubes up to size 16. A lower proportion of cubies to be set without macros would apply if the centre cubie placement method described in Appendix B is used.

Table 2: Total number of cubies and number of cubies to be set without macros

Cube size	Total number of cubies	Cubies to be set without macros		
		Number	% of total	
2	8	0	0.00	
3	26	10	38.46	
4	56	20	35.71	
5	98	38	38.78	
6	152	64	42.11	
7	218	90	41.28	
8	296	132	44.59	
9	386	166	43.01	
10	488	224	45.90	
11	602	266	44.19	
12	728	340	46.70	
13	866	390	45.03	
14	1016	480	47.24	
15	1178	538	45.67	
16	1352	644	47.63	
			Average	40.42

5. Placement of Bottom Layer Centre and Edge Cubies

The first step in solving the cube is to align the lower layer, excluding the corner cubies. For the size 2 cube that means doing nothing at this step (proceed to Section 8 in that case). For the size 3 cube all that is involved is the placing of the edge cubies around a single centre cubie of matching colour. For cubes of size 4 and above, place the centre cubies first, then the edge cubies. A typical approach for the centre cubies would be to gather the required elements one column at a time on the front face and then move the column down onto the bottom face. Starting from a central column and moving outwards would be the normal approach for the centre cubies. Alternatively, centre cubie macros can be used to facilitate bottom layer alignment as described in Appendix B. When all centre cubies are in place, the edge cubies can be aligned one edge set at a time. Normally, the sets would be aligned on the front face and moved into position on the bottom layer using F, -F, L or -R as appropriate. Make sure the exact face colour sequence for left, front, right and back is followed for the edge cubie sets. No macros are defined for this step. However, macro Q that inverts the edge set on the left side of the front face may facilitate the alignment of the bottom layer edge cubies.

When scrolling is active for the Java version of the Unravel program, it is recommended that the bottom face centre cubies be aligned on the front face before moving that face down.

6. Placement of Remaining Lower Centre Cubies

observe that cubie 4 on the left face is in the correct relative location to be shifted to position 4 on the column of centre cubies being aligned. Use $-#4U$ to shift this cubie into position. Use F to place the set of centre cubies being aligned horizontally on the up-side of the front face (blue digits 1-?-3-4-?-6 but black digits 2 and 5 need to be swapped). Observe that red cubie item 6 on the right face is now in the correct relative position for movement into the set being aligned. Use $#3U-F#2U-#3U-F-#2UF$ to put cubie item 6 in position and return set being aligned to horizontal position. Note that cubies on the right side of centre need to be transferred to the left side of the front face (via $-F$) whereas, when applicable, any on the left side would need to be transferred to the right side of the front face (via F in lieu of $-F$). That leaves just cubie 2 to be aligned. Observe that cubie 2 is available from the upper layer. Use $#6LF-#6L-F$ to move cubie 2 into position. If rotation of the upper layer, to put cubie 2 in position, resulted in the red face bottom layer edge cubies being located on the upper (back) edge, use $#6LUF-U-#6L-F$ in lieu of above. Don't transfer from the upper face with the red face bottom layer edge cubies located on the lower (front) edge. In that case rotate the upper layer + or - 90 degrees, as applicable, so that the required cubie is still located on the upper half of the top layer, and in a position to match the set being aligned when placed in a vertical position (left or right of front face), but shift the set being aligned sideways using $#3U$ before moving the cubie down from upper layer.

A practical example for a size 8 cube is considered below. The intention is to initially align the centre cubies in the 7th layer from the top (2nd from bottom), followed by the 6th and 5th layers from the top. For the example illustrated in Fig. 2, the alignment of the last face for the 5th layer is considered as shown in Fig. 2a. The same technique can be used for the other faces for the 5th layer and for all four faces for the 6th and 7th layers.

Enter $FF3U$ to clear the centre face as a work area as shown in Fig. 2b. In this particular case enter $F#3UF-#3UF$ to place the strip of cubies to be set vertically on the right side of front face and move cubie from right face to fill the position on the third row from the top (Fig. 2c). Locate strip of cubies to be set vertically on the left side. Now move cubie from right face into position on second row from top using $#2U$ followed by F to locate the strip of cubies to be set horizontally on the fourth row from the top (Fig. 2d). It can be observed that one cubie on the right face matches a location on the strip that needs to be set. Use $#4UF#2U-#4UF-#2U$ to move that cubie into position (Fig. 2e). Using FF it is observed that the final cubie to be set is available on the upper face (Fig. 2f). Use sequence $F#4U#4LF-#4L-#4UF#4U$ to transfer that cubie into position (Fig. 2g): The full set of required centre cubies is now available and is placed into final position using $-F-3UFF$ (Fig. 2h).

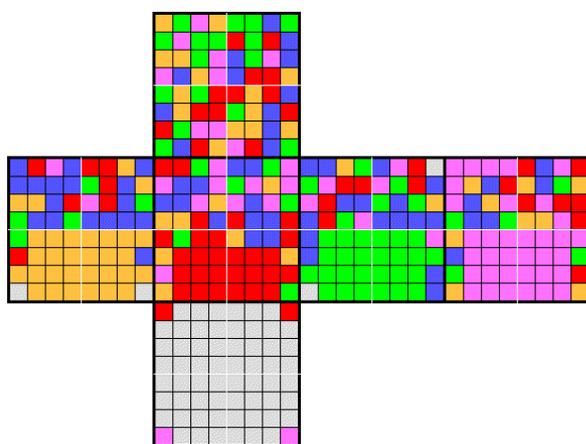


Figure 2a
Initial state

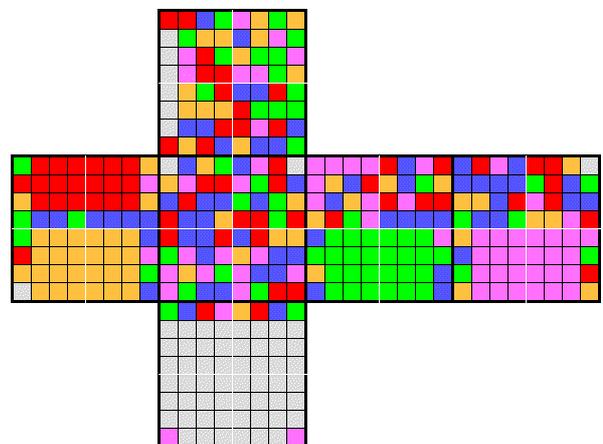


Figure 2b
After FF3U

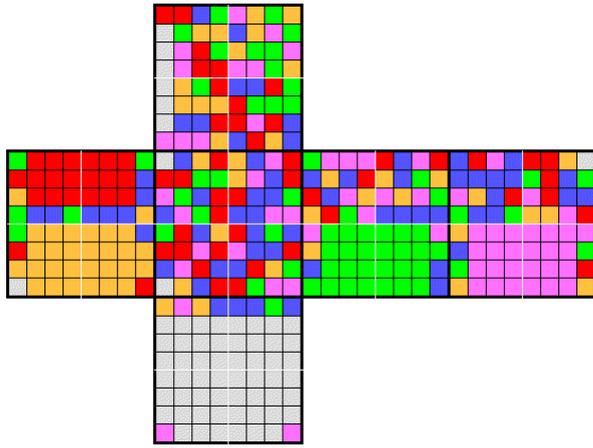


Figure 2c
After F#3UF-#3UF

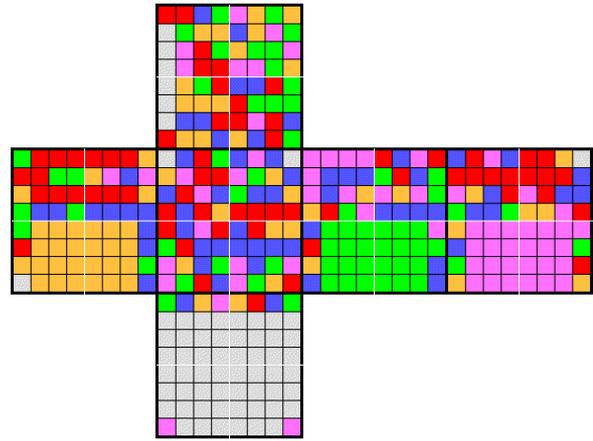


Figure 2d
After #2UF

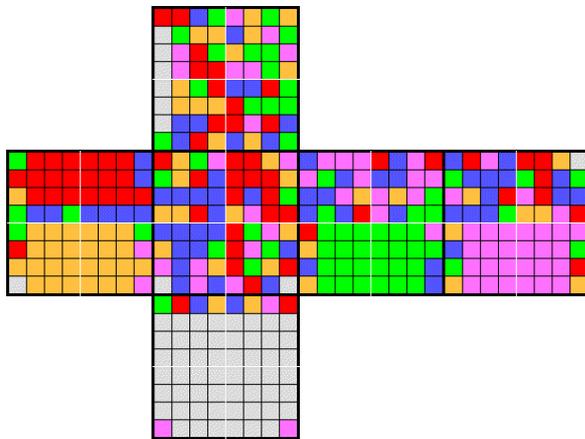


Figure 2e
After #4UF#2U-#4UF-#2U

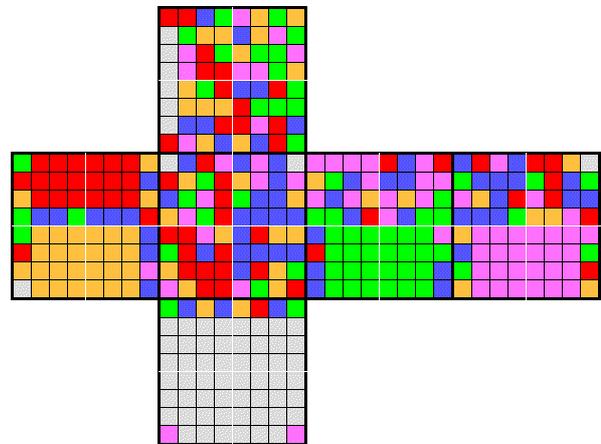


Figure 2f
After FF

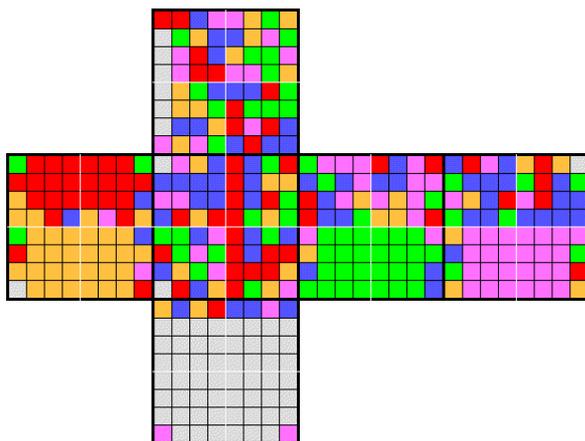


Figure 2g
After F#4U#4LF-#4L-#4UF#4U

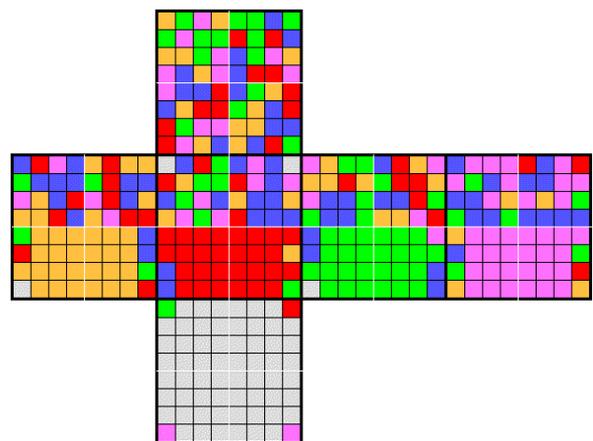


Figure 2h
After -F-3UFF

7. Placement of Upper Centre Cubies

If the method described above in Sec. 6 is used to place centre cubies below the centre line of the cube, then only those above or on the centre line remain to be placed. However, default macros are available for placing all centre cubies, and can be used (Appendix B) if desired. A different macro is provided for each centre cubie location and because the number of centre

cubies requiring placement is a function of the square of the cube size, the number of macros provided for this item far outnumber the remaining macros provided for solving the cube as cube size becomes large (above size 9).

Macro number references used in this document are based on an origin of (row 1, column 1) being the location of the top left corner cubie on the front face. An optional origin of (row 0, column 0) can be set for the Unravel program.

For a cube of size n the number m of macros provided is given by:

$$m = (n - 2)^2 \quad \text{for } n \text{ even.}$$

$$m = (n - 2)^2 - 1 \quad \text{for } n \text{ odd.}$$

Numerical values of m as a function of cube size for cubes up to size 16 are given below.

n	4	5	6	7	8	9	10	11	12	13	14	15	16
m	4	8	16	24	36	48	64	80	100	120	144	168	196

A simple algorithm that results in a transfer between centre cubies in equivalent positions on the front and upper faces is used:

$$\langle rc \rangle = \#cB\#rU - \#cBU\#cB - \#rU - \#cBUU$$

where r is row number and c is column number.

This algorithm does not alter the alignment of any edge cubie sets or corner cubies. Centre cubie placement can be done in any order. Application of $\langle rc \rangle \langle rc \rangle UU$ results in no change of cube state.

For odd size cubes the absolute centre cubie's position is fixed so no macro is needed for its placement. The macros would be used to shift desired cubies to the upper face from any of the initially left, front, right or back faces (after shifting them to the front face) and from the upper face to the face being set after locating it at the front face. Macro K that rotates all but the upper face about the down D face axis may be useful.

The number references for macros used for the transfer of centre cubies from equivalent upper face locations to front face locations for size 8 and size 9 cubes are indicated in Figs. 3a and 3b respectively. The macro number references conveniently follow a row-column coordinate arrangement. Before the macros can be used for final placement it is necessary to transfer the required cubie to the upper face (if it is not already there) and then ensure it is in the correct position on the upper face (use U, -U or UU to achieve this). In most cases it will be necessary to first use one of the centre cubie macros to transfer the required cubie from the front face to the upper face. Initially the required cubie may be located on the left, right or back faces. These faces may be transferred to the front face using K, -K, or KK respectively. If macros are used only for placing the upper centre cubies, there will always be two locations from which one can look for any given cubie. If macros are used for placing all centre cubies there will always be four locations from which one can look for any given cubie. These locations are illustrated in Figs 4a and 4b for size 8 and size 9 cubes respectively. For example, if the transfer of a cubie that would be put into its final position using the M37 macro were required, then one could use the M23 macro to transfer a cubie of the correct face colour to the upper face from an equivalent location on the top row of the centre cubies.

The relevant transfer locations can be visualized by assessing mentally the effect of rotations in 90 degree steps.. Alternatively, the transfer locations can easily be expressed in general terms for cubes of size 4 and above. In general, if a given cubie has coordinates (i, j), then the coordinates of the other locations on the same orbit are as given in the following table.

Reference	Reference + 90°	Reference + 180°	Reference - 90°
-----------	-----------------	------------------	-----------------

(i, j)	(j, n - i + 1)	(n - i + 1, n - j + 1)	(n - j + 1, i)
--------	----------------	------------------------	----------------

where $x = 1$ for (1,1) origin (the default) or $x = -1$ for (0,0) origin.

For example, for a size 15 cube with a (1,1) origin alternative locations to look for a cubie for the (7, 6) location would be locations (6, 9), (9, 10) and (10, 7). If the required cubie were collected from the alternative location (6, 9) macros M69 would be used for the initial transfer and macros M76 would be used for the final placement.

For odd size cubes the rows of cubies bisected by the centre-line of left, front, right and back faces are a special case. The absolute centre cubie resides on a fixed frame and defines the colour of the face on which it resides.

The alignment of the centre cubies on the top layer proceeds in parallel with the alignment of the centre cubies on the left, front, right and back layers. Completion of the alignment on the left, front, right and back layers also means the completion of the alignment of the centre cubies on the upper layer.

	22	23	24	25	26	27	
	32	33	34	35	36	37	
	42	43	44	45	46	47	
	52	53	54	55	56	57	
	62	63	64	65	66	67	
	72	73	74	75	76	77	
	x	x	x	x	x	x	

Figure 3a Centre cubie macro numbers for size 8 cube

	22	23	24	25	26	27	28	
	32	33	34	35	36	37	38	
	42	43	44	45	46	47	48	
	52	53	54		56	57	58	
	62	63	64	65	66	67	68	
	72	73	74	75	76	77	78	
	82	83	84	85	86	87	88	
	x	x	x	x	x	x	x	

Figure 3b Centre cubie macro numbers for size 9 cube

Note: × refers to cubies that have always been previously set.
Some users may not use macros for the locations shown in red.

	9	8	7	3	6	9	
	6	5	4	2	5	8	
	3	2	1	1	4	7	
	7	4	1	1	2	3	
	8	5	2	4	5	6	
	9	6	3	7	8	9	
	x	x	x	x	x	x	

Figure 4a Centre cubie 90 degree rotation for size 8 cube

	12	11	10	3	6	9	12	
	9	8	7	2	5	8	11	
	6	5	4	1	4	7	10	
	3	2	1	0	1	2	3	
	10	7	4	1	4	5	6	
	11	8	5	2	7	8	9	
	12	9	6	3	10	11	12	
	x	x	x	x	x	x	x	

Figure 4b Centre cubie 90 degree rotation for size 9 cube

Note: × refers to cubies that have always been previously set.

Unlike the corner and edge cubies, which can have only one correct location for a solved cube, the centre cubies can end up in different locations each time the cube is solved. Each centre cubie, except the absolute centre cubie for odd size cubes of size 5 and above, can occupy any one of four locations and still appear to be correctly positioned from the perspective of centre

cubies. However, some final layer alignment issues that arise with edge cubies can only be resolved with some further manipulation of centre cubies. These issues comprise:

- Possible need to flip some elements of the final edge cubie set, using the Final Edge Set Alignment Macro(s), for odd and even size cubes of size 4 and above.
- Possible need to flip an odd number of final layer edge cubie sets, using the Single Edge Set Flip Macro for Final Layer Edge Cubes, for even size cubes of size 4 and above (can also be resolved with the Final Edge Set Alignment Macro(s)).
- Possible need to swap the locations of a single pair of final layer edge cubie sets, using the Final Layer Diametrically Opposite Edge Set Swap Macro, for even size cubes of size 4 and above.

8. Placement of Bottom Layer Corner Cubies

Macros A, C and E are provided for transferring the upper left corner cubie on the front face to the lower left corner cubie on the front face (which is the front left corner cubie on the bottom face). These macros apply to cubes of all sizes. Use macro A if the colour of the front face of the upper cubie is the same as that for the bottom face of the cube, C if the colour of the side (left) face of the upper cubie is the same as that for the bottom face of the cube, and E if the colour of the top face of the upper cubie is the same as that for the bottom face of the cube.

$$\begin{aligned} A &= FU-F \\ C &= -L-UL \\ E &= -LULUUC \end{aligned}$$

These corner cubies can be placed after the bottom layer edge cubies are placed if desired.

9. Alignment of Remaining Eight Edge Cubie Sets

In the context of edge cubie sets "alignment", as used here, means two things:

- All cubies in each set must belong on the same edge of the cube (i.e. only two colours appear in a given set).
- All cubies in each set must have the same colour on corresponding surfaces (i.e. none in a given set needs to be flipped).

Alignment does not have any meaning for the size 3 cube and, for cubes of larger size, the first alignment requirement can always be met using techniques described in this section. The second alignment requirement is always met for the size 4 cube but, for cubes of size 5 and above, the second requirement may not be met using only techniques described in this section. However, all but one edge cubie set can always be aligned using techniques described in this section.

The approach here is to repeatedly use P or $-P$ macros to transfer edge sets between the front left and the upper layer. For the size 5 cube, these macros are used in conjunction with a selection of $\#2U$, $-\#2U$, $\#3U$, $-\#3U$, $\#4U$ and $-\#4U$ to progressively align each edge set. For cubes of size n , up to $\#(n-1)U$ would be used. Macro Q inverts the front left edge set and can also be useful.

Expansions for macros P and Q are as follows:

$$\begin{aligned} P &= FUURU-RUU-F \\ Q &= P-UP \end{aligned}$$

While alignment can proceed in any order there may be a small advantage in leaving the four edge sets that belong on the upper (final) layer till last. The first four sets can be easily aligned and parked on the upper layer when their alignment is completed. Centre cubie rows can be out of position when the transfer to the upper layer occurs for the first four edge sets.

The author's preference for aligning the first four edge sets is to assemble the edge cubies on the left back edge using the following steps:

- Select an initial orientation and move any of the required edge cubies having opposite orientation back to the left front edge, invert with macro Q and transfer cubies back to the selected edge.
- Transfer all required edge cubies of the correct orientation in the other three edges in the mid-section to the selected edge set.
- Invert the selected edge set and move all the remaining members in the other three edges in the mid-section to the selected edge.
- Move each unaligned edge set on the upper face in turn to the left front and do direct or inverted (after applying macro Q) transfer to the selected edge.

If centre cubies have been aligned before the edge cubies as indicated in Sec. 7, then alignment steps for the last four edge sets must not disturb the alignment of centre cubies and of any previously aligned edge set. If centre cubies are to be aligned after the edge cubies (Appendix B) then edge set 5 can be simply aligned. The steps below apply for edge sets 5, 6 and 7 when centre cubies are aligned before edge sets and for edge sets 6 and 7 when centre cubies are aligned after edge sets. The edge set 8 arrangement is what results after edge set 7 is aligned.

Place two adjacent edge sets in position on each side of the selected face (consider this to be violet – the back face colour of the initial set cube for the Unravel program). If centre cubies are aligned, these edge sets must match the violet face colour. If centre cubies are to be aligned later then:

- if the cube size is odd, then the violet centre cubie defines the required face, and
- if the cube size is even, then the violet face can be arbitrarily selected but once selected the violet-red and orange-green axes will now be defined.

For the edge cubie alignment method described here it is assumed that the edge set to be currently aligned is placed at the front (either left side or right side can be selected). If the left side is selected, then any cubie belonging to that set that resides on the right side of the front face needs to be moved across to the left side. For odd size cubes the central edge cubie defines the edge set colours.

When fifth and sixth edge sets are to be aligned, more than one edge set will hold the required cubies. . Alignment of the sixth set deserves special mention. For cubes of large size it would be normal for each of the remaining three unaligned edge sets to contain three different edge cubies. One of the unaligned sets should be placed on the upper face and the other two on the left and right sides of the front face. All cubies for the sixth set should be placed on one side of the front face and the other set on the front face should be swapped with the one on the top face that still has one or more edge cubies for the sixth set being aligned.. The sixth set can now be fully aligned and swapped with the remaining unaligned set on the top face.

When moving the required edge cubies to the front left edge, three possibilities as indicated in the following table can arise.

Possibility	For edge set	Action (r = row number)
-------------	--------------	-------------------------

One edge cubie needs to be moved but no matching cubie exists on this row or its complement on the left side.	5 and 6	Use $\#rUQ-\#rU$ to do transfer.
A complementary pair of cubies need to be moved to the left side.	5, 6 and 7	Use $\#rUQ-\#rU$ to transfer just one. Perform action as described below for the second transfer. Alternatively, $\#rU\#(n-r)UQ-\#(n-r)U$ will do the full transfer where n is cube size.
One edge cubie needs to be shifted from right to left and its complement already resides in the left edge set	5, 6, and 7	Place edge cubie to be shifted on same row as the one already placed on the left. Use $\#rUQ-\#rU$ to do transfer.

For cubes of large size it is likely that some complementary pairs in the edge set being aligned will need to swap positions (a wrong position implies a wrong orientation). Use $\#r\#(n-r)Q-\#r-\#(n-r)$ to correct a complementary pair for edge sets 5, 6 and 7.

Some simple examples of edge set 7 alignment are considered below. In most cases, particularly for cubes of large size, edge set 8 will need a further alignment step (Sec. 10) that requires a rearrangement of centre cubies in the upper face.

In the first example (Example 1), alignment of all edge sets except the last two has been achieved and the remaining non-aligned pair has been located at the left and right sides of the front face as shown in Fig. 5a. Notice that each side has one edge cubie that belongs on the other. After inverting the right side as shown in Fig. 5b, the 2nd and 4th rows will have two edge cubies belonging to one side on the 2nd row and two belonging to the other side on the 4th row. By rotating the 2nd row 90 degrees CW, then inverting the left side front edge cubies, and then rotating the second row back 90 degrees CCW, the state shown in Fig. 5c results. Notice that the middle row edge cubies are flipped relative to their companions on either side. Doing a rotation, inversion and return as above results in perfect alignment, as shown in Fig. 5d, of both edge cubie sets. In this example, these last two edge cubie sets belong on the upper layer. Simple swapping of these with the relevant previously aligned edge sets located on the upper face, using P or $-P$ macros completely restores everything below the top layer.

In the second example (Example 2), alignment of all edge sets except the last two has been achieved and the remaining non-aligned last pair has been located at the left and right sides of the front face as shown in Fig. 6a. Each side has one edge cubie that belongs on the other side. Proceed, as for Fig. 5b and Fig. 5c in Example 1, to the states shown in Fig. 6b and Fig. 6c. Only one (Fig. 6c) of the last two sets is perfectly aligned. The simple techniques referred to in this section cannot resolve this. Place the non-aligned edge set at the front of the upper layer (Fig. 6d). For the size 5 cube the Final Edge Set Alignment macro M12 for numerical macro origin (1,1), as described for the Final Edge Set Alignment Macro(s) in the next section and in the Unravel program on-line Macro Help, will fix the problem.

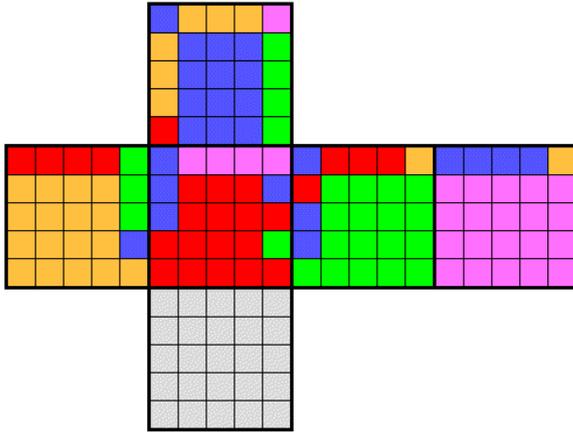


Figure 5a
Example 1: Initial state

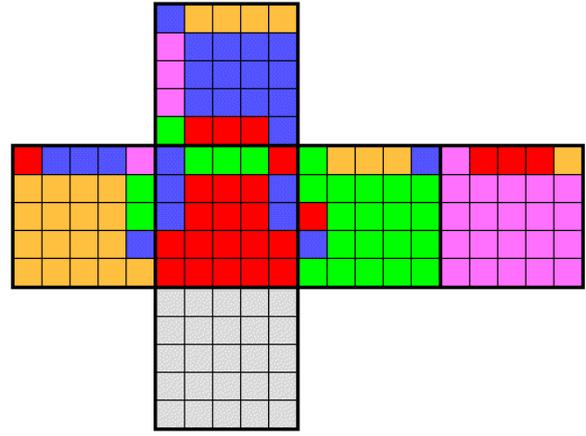


Figure 5b
Example 1: After WUQ-WU

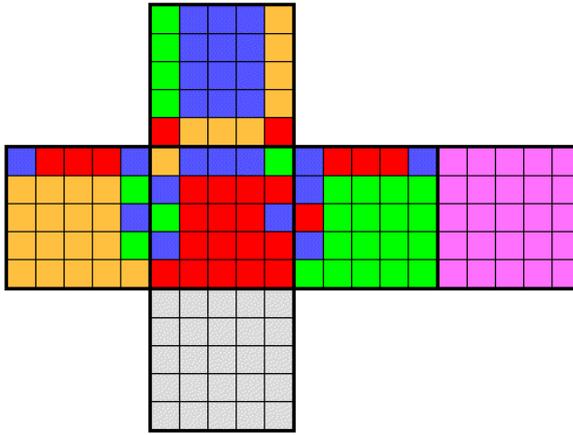


Figure 5c
Example 1: After #2UQ-#2U

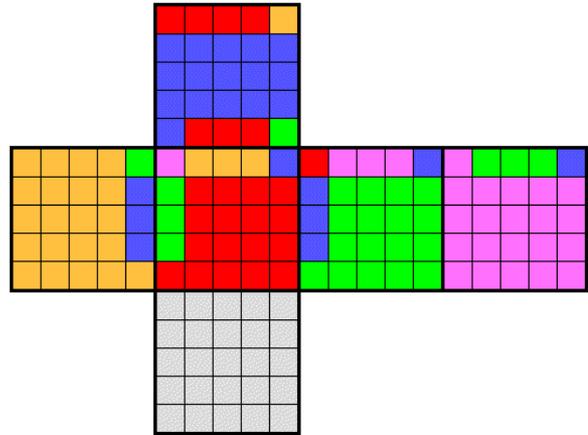


Figure 5d
Example 1: After #3UQ-#3U

The techniques described in this section can be used for cubes of size 3 and above but simplifications arise for the size 3 cube. The procedures illustrated in Examples 1 and 2 are also applicable to cubes of size 6 and above. The # n UQ-# n U sequence would normally have to be applied for multiple rows as the cube size is increased. When aligning the last pair of edge cube sets, the approach would be to get one of the pair aligned properly and take what comes for the other (which is what was done for the size 5 cube in Example 2).

The examples for the size 5 cube may be too simple to adequately illustrate the technique for cubes of larger size. Table 3 provides an example of the technique that was used to align one of the last pair of edge cube sets for a size 15 cube. Elements of the first set are designated ab (ba when flipped) and those of the second set as cd (dc when flipped). The initial arrangement is shown at the left in Table 3. Since this is an odd size cube, each edge cubie set has a central (row 8) cubie whose position cannot be moved up or down. Observe that the central cubie on the left is ab and that on the right is dc , so we will aim to put the ab elements on the left and the cd elements on the right. The approach is to firstly place all elements of the ab set (irrespective of whether some may need to be flipped) at the left and all elements of the cd set on the right. Observe that there are initially five cd elements on the left. The condition we look for is a cd (or dc) element that has a vertical mirror image ab (or ba) cubie on the left with its flipped counterpart on the right as illustrated in Table 4 where n is cube size (15 for this example).

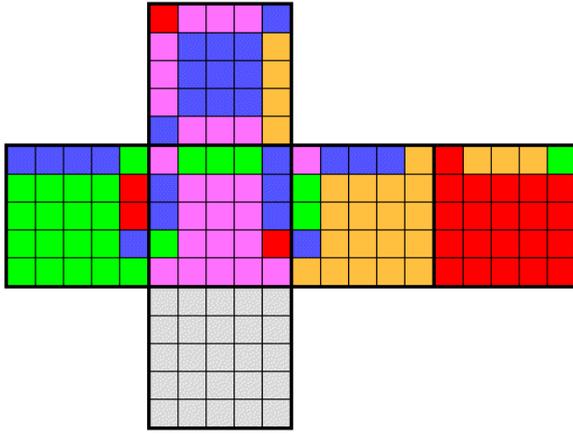


Figure 6a
Example 2: Initial state

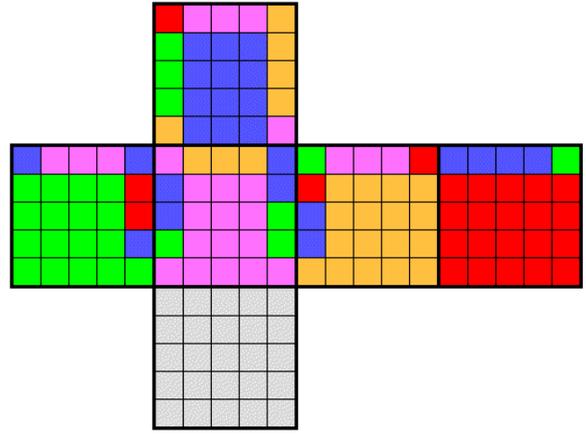


Figure 6b
Example 2: After WUQ-U

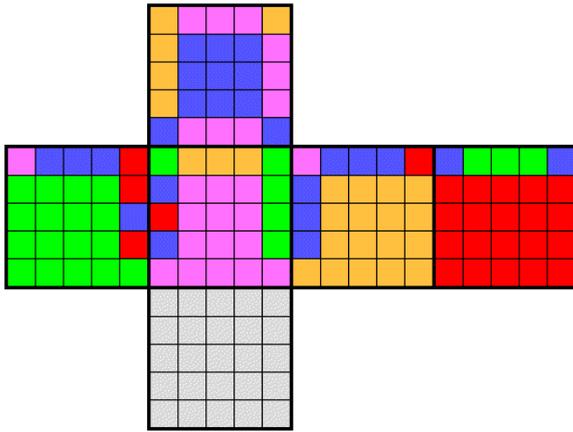


Figure 6c
Example 2: After #2UQ-#2U

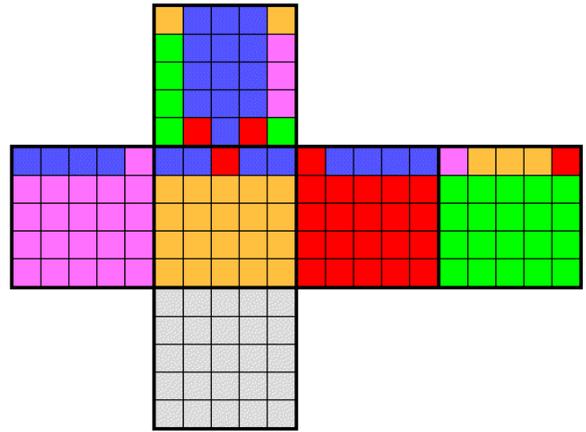


Figure 6d
Example 2: After UU-PWU-U-PUU

Table 3: Example of alignment of one of the last pair of edge cubie sets for a size 15 cube

Row	Initial		After step 1		After step 2		After step 3	
	LF	RF	LF	RF	LF	RF	LF	RF
1								
2	ab	cd	dc	cd	ba	cd	ba	dc
3	ba	ab	ba	dc	ab	dc	ba	dc
4	ab	ba	ab	dc	ba	dc	ba	cd
5	ab	cd	ba	cd	ab	cd	ba	cd
6	dc	cd	ba	cd	ab	cd	ba	cd
7	cd	ab	ba	ab	ba	cd	ba	dc
8	ab	dc	ba	dc	ab	dc	ba	dc
9	ab	cd	dc	cd	ba	cd	ba	dc
10	ba	ab	ba	cd	ab	cd	ba	cd
11	ab	cd	ba	cd	ab	cd	ba	cd
12	cd	dc	ab	dc	ba	dc	ba	cd
13	cd	dc	ba	dc	ab	dc	ba	dc
14	cd	ab	ba	ab	ba	cd	ba	dc
15								

LF Left side Front face RF Right side Front face

Table 4: Technique for aligning second last edge cube set

Row	Left Front	Right Front	Action
k	cd (or dc)		
$n - k + E$	ab (or ba)	ba (or ab)	Move this row CW with $\#(n - k + E)U$

where $E = n \bmod 2$ (equal to 0 for n even, equal to 1 for n odd)

Observe in Table 4 that the condition is met for rows (k values) 6, 12 and 13, so rows 10, 4 and 3 need to be rotated 90 degrees CW about the upper face axis, then the left column needs to be inverted and rows 10, 4 and 3 need to be rotated 90 degrees CCW as indicated in the Step 1 sequence. That leaves two remaining cd elements in the left column in rows 2 and 9. These can be moved by rotating rows 14 and 7 CW and proceeding as above using the Step 2 sequence. All the ab elements are now in the left column but some (rows 2, 4, 7, 9, 12 and 14) need to be flipped. That is simply done with the Step 3 sequence. All ab elements are now properly aligned. As expected, some of the cd elements need to be flipped. That can be achieved using the Final Edge Set Alignment Macros described in the next section and in the Unravel program on-line macro help.

Step 1 sequence #3U#4U#10UQ-#10U-#4U-#3U

Step 2 sequence #7U#14UQ-#14U-#7U

Step 3 sequence #2U#4U#7U#9U#12U#14UQ-#14U-#12U-#9U-#7U-#4U-#2U

10. Final Layer Actions

Having reached the final layer, the further steps required are readily definable and are perfect candidates for macro solutions. Macros are available for all actions required in the final layer. Difficulties may arise if actions are not performed in the sequence shown in the Unravel program on-line macro help and reproduced in Table 5.

Table 5: List of Final Layer Macros

Order	Macros	Comment
1	Final Edge Set Alignment Macro(s)	Need can arise only for cubes of size 4 and above.
2	Single Edge Set Flip Macro for Final Layer Edge Cubies	Need can arise only for even-size cubes of size 4 and above. Alternatively, Macros for previous item can be used to achieve the required result.
3	Final Layer Corner Location Macros	Need can arise for cubes of all sizes.
4	Final Layer Edge Cubie Location Macros	Need can arise only for cubes of size 3 and above.
5	Final Layer Diametrically Opposite Edge Set Swap Macro	Need can arise only for even-size cubes of size 4 and above.
6	Final Layer Corner Twirl Macro	Need can arise for cubes of all sizes.
7	Final Layer Edge Set Flip Macro	Need can arise only for cubes of size 3 and above.

Macros as indicated in Order 1, 2 and 5 are used to resolve issues that cannot be solved using standard cube size 3 algorithms. These issues are expanded upon in Appendix C and Appendix D. Macro details, as provided in the on-line macro help, have been reproduced below. Where the number or content of macros varies according to cube size, only details applicable to a size 16 cube are shown here.

1. Final Edge Set Alignment Macros

Some adjustment of the final edge set may be required before all the final layer edge cubes achieve the same sense. Place the final edge cube set to be aligned at the front edge of the upper layer. Adjustment can proceed in any order until all cubes in the final edge set are aligned in the same sense.

The general form for the final edge set alignment macros, for a (1,1) numerical macro origin, is:

$$\langle 1c \rangle = \#cR\#cRBBUU\#cLUU-\#cRUU\#cRUUFF\#cRFF-\#cLBB\#cR\#cR$$

where c is the column number of the left cubie of the pair to be flipped.

There is no cube-size-dependent item in this algorithm. The only effect of cube size is the need for more macros of this form.

For the size 16 cube the following apply.

```

<18> converts [- - - - - b b - - - - -] to [- - - - - a a - - - - -].
<17> converts [- - - - - b - - b - - - - -] to [- - - - - a - - a - - - - -].
<16> converts [- - - - b - - - - b - - - -] to [- - - - a - - - - a - - - -].
<15> converts [- - - b - - - - - b - - -] to [- - - a - - - - - a - - -].
<14> converts [- - b - - - - - - - b - -] to [- - a - - - - - - - a - -].
<13> converts [- b - - - - - - - - - b -] to [- a - - - - - - - - - a -].
<12> converts [b - - - - - - - - - - - b] to [a - - - - - - - - - - - a].

```

Note: "a" and "b" refer to the two faces of the edge cubes.

"-" means "don't care" (i.e. either "a" or "b").

```

<12> = #2R#2RBBUU#2LUU-#2RUU#2RUUFF#2RFF-#2LBB#2R#2R
<13> = #3R#3RBBUU#3LUU-#3RUU#3RUUFF#3RFF-#3LBB#3R#3R
<14> = #4R#4RBBUU#4LUU-#4RUU#4RUUFF#4RFF-#4LBB#4R#4R
<15> = #5R#5RBBUU#5LUU-#5RUU#5RUUFF#5RFF-#5LBB#5R#5R
<16> = #6R#6RBBUU#6LUU-#6RUU#6RUUFF#6RFF-#6LBB#6R#6R
<17> = #7R#7RBBUU#7LUU-#7RUU#7RUUFF#7RFF-#7LBB#7R#7R
<18> = #8R#8RBBUU#8LUU-#8RUU#8RUUFF#8RFF-#8LBB#8R#8R

```

2. Single Edge Set Flip Macro for Final Layer Edge Cubies

Before proceeding further with the final layer alignment it is essential that the number of edge sets that need to be flipped is not odd (one or three). If an odd number of matched edge sets needs to be flipped on the final layer (can only occur for even size cubes), use S to flip the set located at the front of the upper layer. The general form of the edge set flip macro S is:

$$S = mRmRRRBBUUmL-LUU-mRRUUmR-RUUFFmR-RFF-mLLBBmRmRRR$$

where $m = n/2$ and n is cube size.

For the size 16 cube the following applies.

$$S = 8R8RRRBBUU8L-LUU-8RRUU8R-RUUFF8R-RFF-8LLBB8R8RRR$$

Alternatively, the required result can be achieved using the "Final Edge Set Alignment Macro(s)".

3. Final Layer Corner Location Macros

Locate final layer at top and use T to swap the two front corners.

$$T = B-U-FU-B-UFUU$$

Sequences containing T, U and -U can be used to cover all other final layer corner location possibilities not covered by the use of a single T macro. Such sequences together with alternative macro definitions are as follows:

With lower right corner cubie of upper face fixed, use UUTUTU or G to provide CW movement of three remaining corner cubies.

With lower right corner cubie of upper face fixed, use -UT-UTUU or -G to provide CCW movement of three remaining corner cubies.

Use TUT-UT or H to provide diagonal swap of lower left with upper right corner cubie.

$$G = UUTUTU$$

$$H = TUT-UT$$

4. Final Layer Edge Cubie Location Macros

Locate final layer at top. Use V to obtain a CW movement, and -V to provide a CCW movement, of the leftmost three edge cubies (or cubie sets).

$$V = LLU-FBLL-BFULL$$

Sequences containing V, U and -U can be used to cover other final layer edge cubie (or edge cubie set) location possibilities not covered by the use of a single V or -V macro. Such sequences together with alternative macro definitions are as follows:

Swap left with right, and up with down edge cubies (or cubie sets) using V-UVU or I.

Swap left with down, and up with right cubies (or cubie sets) using -V-U-VU or J.

$$I = V-UVU$$

$$J = -V-U-VU$$

Even size cubes above size 2 may require the swapping of just one opposite pair of edge cubie sets. Refer to "Final Layer Diametrically Opposite Edge Set Swap Macro".

5. Final Layer Diametrically Opposite Edge Set Swap Macro

Locate final layer at top. Locate one of the sets to be swapped at the front and use X to exchange it with the edge set at the back. The general form of the diametrically opposite edge set swap macro X is:

$$X = mRmRRRUUmRmRRRUUmUmUUUmRmRRRmUmUUU$$

where $m = n/2$ and n is cube size.

For the size 16 cube the following applies:

$$X = 8R8RRRUU8R8RRRUU8U8UUU8R8RRR8U8UUU$$

Alternatively, the required rearrangement can be considered as being the combined effect of a number of moves involving just four edge cubies^[9] as for the size 4 cube. That approach has not been used for the instructions detailed in this document.

6. Final Layer Corner Twirl Macro

Locate final layer at the top. Use Y to provide a CW twirl of the front right corner. Locate corner requiring a CCW twirl at the front right corner and use -Y.

$$Y = -RDRFD-F$$

7. Final Layer Edge Set Flip Macro

Locate final layer at the top. Use Z to flip edge cubie (or cubie set) located at the front. Locate other top layer edge cubie (or cubie set) requiring a flip at the right side and use -Z.

$$Z = FU-DLLJUDDR$$

11. Centre Cubie Block Transfer Macro

For users, who sequentially align all centre cubies on a face before doing the same on the next face, the ability to transfer all centre cubies, irrespective of facelet colour, from the front to the upper face can be useful, particularly for cubes of large size and for the early stage of the centre cubie alignment process (up to the fourth face only).

The macro that performs the transfer involves a rearrangement of the cubies on these faces and that is of no consequence as this transfer would precede alignment using macros listed in the "Centre Cubie Location Macros" section of Appendix A. Except for the centres cubies on the front and upper faces, no change in the alignment of any other cubie occurs. No transfer of the absolute centre cubies for odd size cubes occurs.

For cube size 4 macro <13> performs the transfer and for all cubes of larger size macro <40> (or <0400> for cube of size greater than 16) does the transfer. These cube names are the same for the optional origins [0,0] and [1,1]. For cubes of size greater than 4 the transfer macro makes use of one auxiliary macro for cubes of even size and three auxiliary macros for cubes of odd size.

For the size 4 cube only one transfer macro as defined below for origin [1,1] applies. Replace <22> with <11> when origin [0,0] applies.

$$* \text{ <13> } = \text{ <22>UFU<22>FU<22>FUU<22>-UF}$$

For cubes of size greater than 4, the transfer and auxiliary macros defined below are the same for the [0,0] and [1,1] origins.

Even cubes of size 6 and above

The auxiliary macro names vary with cube size. Define <0a> as the single auxiliary macro required for cubes of even size. In that case for a cube of size n, a = n/2. Define b = n - 1. For cubes of size greater than 16, 4-digit macros apply and in such cases replace 0a with 00aa where aa = n/2 and any leading zero is preserved.

- * <0a> = -BbBaU-UB-bBU-BbB-aUUB-bBUU
- * <40> = <0a>U<0a>FFUU<0a>U<0a>-UUFFU

For a size 16 cube:

- * <08> = -B15B8U-UB-15BU-B15B-8UUB-15BUU
- * <40> = <08>U<08>FFUU<08>U<08>-UUFFU

For a size 40 cube:

- * <0020> = -B39B20U-UB-39BU-B39B-20UUB-39BUU
- * <0400> = <0020>U<0020>FFUU<0020>U<0020>-UUFFU

Odd cubes of size 5 and above

The auxiliary macro names vary with cube size. Define <0a>, <0b> and <0c> as the three auxiliary macros required for cubes of odd size. In that case for a cube of size n:

$a = (n - 1)/2$, $b = a + 1$, $c = a + 2$, and $d = n - 1$.

For cubes of size greater than 16, 4-digit macros apply and in such cases replace 0a with 00aa where aa = $(n-1)/2$ and any leading zero is preserved.

- * <0a> = -BdBaU-UB-dBU-BdB-aUUB-dB-U
- * <0b> = <0a>U<0a>FF<0a><0a>FF-U
- * <0c> = -BaB#bUB-aBU-BaB-#bUB-aB-U
- * <40> = <0b>U<0c>FFUU<0c>UUFF-U

For size 15

- * <07> = -B14B7U-UB-14BU-B14B-7UUB-14B-U
- * <08> = <07>U<07>FF<07><07>FF-U
- * <09> = -B7B#8UB-7BU-B7B-#8UB-7B-U
- * <40> = <08>U<09>FFUU<09>UUFF-U

For size 39

- * <0019> = -B38B19U-UB-38BU-B38B-19UUB-38B-U
- * <0020> = <0019>U<0019>FF<0019><0019>FF-U
- * <0021> = -B19B#20UB-19BU-B19B-#20UB-19B-U
- * <0400> = <0020>U<0021>FFUU<0021>UUFF-U

References

1. Wikipedia, the free web encyclopaedia, Cartesian coordinate system, http://en.wikipedia.org/wiki/Cartesian_coordinate_system, Retrieved 2017-02-12.
2. Wikipedia, the free web encyclopaedia, Permutation, <http://en.wikipedia.org/wiki/Permutation>, Retrieved 2017-02-12.
3. Tom Davis, Permutation Groups, 2 April 2003. <http://www.geometer.org/mathcircles/perm.pdf>.
4. Tom Davis, The Mathematics of the Rubik's Cube: Introduction to Group Theory and Permutation Puzzles, 17 March 2009, <http://web.mit.edu/sp.268/www/rubik.pdf>.
5. Arfur Dogfrey, The Dog School of Mathematics, Introduction to Group Theory, <http://dogschool.tripod.com/index.html>, Retrieved 2017-02-12.
6. Ryan Heise, Rubik's Cube Theory – Parity, <http://www.ryanheise.com/cube/parity.html>, Retrieved 2017-02-12.
7. Fraser, K., UnravelJ - Size 2x2x2 to 99x99x99 for standard cube, size 3x3x3 to 32x32x32 for centres with numerical markings and up to 99x99x99 for centres with corner markings, 2D, Java applet, Java Web Start or Java archive direct download, all platforms. <http://kenblackbox.com/unravelcube.htm>.
8. Fraser, K., Implementing and Solving Rubik's Family Cubes with Marked Centres. <http://kenblackbox.com/cube/solving/markctr.pdf>
9. Fraser, K., Rules for Rubik's Family Cubes of All Sizes, <http://kenblackbox.com/cube/maths/cuberules.pdf>.
10. Ryan Heise, Rubik's Cube Theory - Laws of the cube, http://www.ryanheise.com/cube/cube_laws.html, Retrieved 2017-02-12.
11. Arfur Dogfrey, The Dog School of Mathematics, Rubik's Magic Cube, <http://dogschool.tripod.com/rubikscube.html>, Retrieved 2017-02-12.
12. Fraser, K., Rubik's Cube Extended: Derivation of Number of States for Cubes of Any Size and Values for up to Size 25x25x25. <http://kenblackbox.com/cube/maths/cubestates.pdf>
13. Wikipedia, the free web encyclopaedia, Fridrich Method, http://en.wikipedia.org/wiki/Fridrich_Method, Retrieved 2017-02-12.
14. Chris Hardwick, Fix Parity, <http://www.speedcubing.com/chris/4speedsolve3.html>, Retrieved 2017-02-12.

Appendix A

Algorithms as Defined in Macros for Use with Size 16 Cube

All macros that involve only outer layer movements are the same for cubes of any size. The number, or content, of other macros is dependent on cube size. Default macros can be automatically generated and loaded by the Unravel program at run-time. The following (except for section numbers) is a copy of the macro help text automatically generated when macros for a size 16 cube are loaded. The macro help text for the Java version of Unravel, are loaded. The macro help text for the C++ version is the same except for the "Centre Cubie Block Transfer Macro" section that is absent.

File "urvcmd16.txt" Macro Commands

This file contains macro definitions and instructions on where they are to be used.

1. Section of Cube to be Set Without Macros

The extent to which macros are used is optional and users may decide to not use some or all of them. When macros are used, there are some restrictions on the way the cube is unscrambled.

Align bottom layer centre cubies and then edge cubies. Although placement macros are provided for all centre cubies except those on the bottom layer, the placement of the centre cubies on the lower half of the F, R, B and L faces without using macros may be preferred. Also, the bottom layer centre cubies can be placed using centre cubie macros by initially doing the placement on the front face.

2. Centre Cubie Block Transfer Macro

For users, who sequentially align all centre cubies on a face before doing the same on the next face, the ability to transfer all centre cubies, irrespective of facelet colour, from the front to the upper face can be useful, particularly for cubes of large size and for the early stage of the centre cubie alignment process (up to the fourth face only).

Macro <40> transfers all centre cubies on the front face for this even size cube. The transfer involves a rearrangement of the cubies on these faces and that is of no consequence as this transfer would precede alignment using macros listed in the "Centre Cubie Location Macros" section. Except for the centres cubies on the front and upper faces, no change in the alignment of any other cubie occurs.

* <40> = <08>U<08>FFUU<08>U<08>-UFFU

Auxiliary macro <08> is used in the definition of macro <40>.

* <08> = -B15B8U-UB-15BU-B15B-8UUB-15BUU

3. Centre Cubie Location Macros

Alignment macros are provided for all the centre cubies on the front face, where rows are counted from the top (Row 1) downwards.

Centre cubies can be aligned in any order. No change in the alignment of edge or corner cubies applies when the centre cubie location macros are used (so users can align edge cubies before centre cubies if they wish). Centre cubie macros swap an upper layer cubie with one in the equivalent position on the front layer. No other movement occurs on the front layer but some rearrangement occurs on the upper layer in conformity with an even parity requirement. Centre cubie alignment is a two step process. Firstly, if desired centre cubies are not located on the upper face, a face containing the required centre cubies is placed at the front and centre cubie macros are used to transfer them to the upper face. Secondly, the face being aligned is then placed in the front and centre cubie macros are used to retrieve the required centre cubies that have just been placed on the upper face.

The details are given below.

```

Row 01:  xx   xx   ....  xx   xx
Row 02:  xx  <22>  ....  <2F>  xx
Row 03:  xx  <32>  ....  <3F>  xx
Row 04:  xx  <42>  ....  <4F>  xx
Row 05:  xx  <52>  ....  <5F>  xx
Row 06:  xx  <62>  ....  <6F>  xx
Row 07:  xx  <72>  ....  <7F>  xx
Row 08:  xx  <82>  ....  <8F>  xx
Row 09:  xx  <92>  ....  <9F>  xx
Row 10:  xx  <A2>  ....  <AF>  xx
Row 11:  xx  <B2>  ....  <BF>  xx
Row 12:  xx  <C2>  ....  <CF>  xx
Row 13:  xx  <D2>  ....  <DF>  xx
Row 14:  xx  <E2>  ....  <EF>  xx
Row 15:  xx  <F2>  ....  <FF>  xx
Row 16:  xx   xx   ....  xx   xx

```

xx = no centre cubie macro applicable

```

* <22> = #2B#2U-#2BU#2B-#2U-#2BUU
* <23> = #3B#2U-#3BU#3B-#2U-#3BUU
* <24> = #4B#2U-#4BU#4B-#2U-#4BUU
* <25> = #5B#2U-#5BU#5B-#2U-#5BUU
* <26> = #6B#2U-#6BU#6B-#2U-#6BUU
* <27> = #7B#2U-#7BU#7B-#2U-#7BUU
* <28> = #8B#2U-#8BU#8B-#2U-#8BUU
* <29> = #9B#2U-#9BU#9B-#2U-#9BUU
* <2A> = #10B#2U-#10BU#10B-#2U-#10BUU
* <2B> = #11B#2U-#11BU#11B-#2U-#11BUU
* <2C> = #12B#2U-#12BU#12B-#2U-#12BUU
* <2D> = #13B#2U-#13BU#13B-#2U-#13BUU
* <2E> = #14B#2U-#14BU#14B-#2U-#14BUU
* <2F> = #15B#2U-#15BU#15B-#2U-#15BUU
* <32> = #2B#3U-#2BU#2B-#3U-#2BUU
* <33> = #3B#3U-#3BU#3B-#3U-#3BUU

```

* <34> = #4B#3U-#4BU#4B-#3U-#4BUU
 * <35> = #5B#3U-#5BU#5B-#3U-#5BUU
 * <36> = #6B#3U-#6BU#6B-#3U-#6BUU
 * <37> = #7B#3U-#7BU#7B-#3U-#7BUU
 * <38> = #8B#3U-#8BU#8B-#3U-#8BUU
 * <39> = #9B#3U-#9BU#9B-#3U-#9BUU
 * <3A> = #10B#3U-#10BU#10B-#3U-#10BUU
 * <3B> = #11B#3U-#11BU#11B-#3U-#11BUU
 * <3C> = #12B#3U-#12BU#12B-#3U-#12BUU
 * <3D> = #13B#3U-#13BU#13B-#3U-#13BUU
 * <3E> = #14B#3U-#14BU#14B-#3U-#14BUU
 * <3F> = #15B#3U-#15BU#15B-#3U-#15BUU
 * <42> = #2B#4U-#2BU#2B-#4U-#2BUU
 * <43> = #3B#4U-#3BU#3B-#4U-#3BUU
 * <44> = #4B#4U-#4BU#4B-#4U-#4BUU
 * <45> = #5B#4U-#5BU#5B-#4U-#5BUU
 * <46> = #6B#4U-#6BU#6B-#4U-#6BUU
 * <47> = #7B#4U-#7BU#7B-#4U-#7BUU
 * <48> = #8B#4U-#8BU#8B-#4U-#8BUU
 * <49> = #9B#4U-#9BU#9B-#4U-#9BUU
 * <4A> = #10B#4U-#10BU#10B-#4U-#10BUU
 * <4B> = #11B#4U-#11BU#11B-#4U-#11BUU
 * <4C> = #12B#4U-#12BU#12B-#4U-#12BUU
 * <4D> = #13B#4U-#13BU#13B-#4U-#13BUU
 * <4E> = #14B#4U-#14BU#14B-#4U-#14BUU
 * <4F> = #15B#4U-#15BU#15B-#4U-#15BUU
 * <52> = #2B#5U-#2BU#2B-#5U-#2BUU
 * <53> = #3B#5U-#3BU#3B-#5U-#3BUU
 * <54> = #4B#5U-#4BU#4B-#5U-#4BUU
 * <55> = #5B#5U-#5BU#5B-#5U-#5BUU
 * <56> = #6B#5U-#6BU#6B-#5U-#6BUU
 * <57> = #7B#5U-#7BU#7B-#5U-#7BUU
 * <58> = #8B#5U-#8BU#8B-#5U-#8BUU
 * <59> = #9B#5U-#9BU#9B-#5U-#9BUU
 * <5A> = #10B#5U-#10BU#10B-#5U-#10BUU
 * <5B> = #11B#5U-#11BU#11B-#5U-#11BUU
 * <5C> = #12B#5U-#12BU#12B-#5U-#12BUU
 * <5D> = #13B#5U-#13BU#13B-#5U-#13BUU
 * <5E> = #14B#5U-#14BU#14B-#5U-#14BUU
 * <5F> = #15B#5U-#15BU#15B-#5U-#15BUU
 * <62> = #2B#6U-#2BU#2B-#6U-#2BUU
 * <63> = #3B#6U-#3BU#3B-#6U-#3BUU
 * <64> = #4B#6U-#4BU#4B-#6U-#4BUU
 * <65> = #5B#6U-#5BU#5B-#6U-#5BUU
 * <66> = #6B#6U-#6BU#6B-#6U-#6BUU
 * <67> = #7B#6U-#7BU#7B-#6U-#7BUU
 * <68> = #8B#6U-#8BU#8B-#6U-#8BUU
 * <69> = #9B#6U-#9BU#9B-#6U-#9BUU
 * <6A> = #10B#6U-#10BU#10B-#6U-#10BUU
 * <6B> = #11B#6U-#11BU#11B-#6U-#11BUU
 * <6C> = #12B#6U-#12BU#12B-#6U-#12BUU
 * <6D> = #13B#6U-#13BU#13B-#6U-#13BUU
 * <6E> = #14B#6U-#14BU#14B-#6U-#14BUU
 * <6F> = #15B#6U-#15BU#15B-#6U-#15BUU

* <72> = #2B#7U-#2BU#2B-#7U-#2BUU
 * <73> = #3B#7U-#3BU#3B-#7U-#3BUU
 * <74> = #4B#7U-#4BU#4B-#7U-#4BUU
 * <75> = #5B#7U-#5BU#5B-#7U-#5BUU
 * <76> = #6B#7U-#6BU#6B-#7U-#6BUU
 * <77> = #7B#7U-#7BU#7B-#7U-#7BUU
 * <78> = #8B#7U-#8BU#8B-#7U-#8BUU
 * <79> = #9B#7U-#9BU#9B-#7U-#9BUU
 * <7A> = #10B#7U-#10BU#10B-#7U-#10BUU
 * <7B> = #11B#7U-#11BU#11B-#7U-#11BUU
 * <7C> = #12B#7U-#12BU#12B-#7U-#12BUU
 * <7D> = #13B#7U-#13BU#13B-#7U-#13BUU
 * <7E> = #14B#7U-#14BU#14B-#7U-#14BUU
 * <7F> = #15B#7U-#15BU#15B-#7U-#15BUU
 * <82> = #2B#8U-#2BU#2B-#8U-#2BUU
 * <83> = #3B#8U-#3BU#3B-#8U-#3BUU
 * <84> = #4B#8U-#4BU#4B-#8U-#4BUU
 * <85> = #5B#8U-#5BU#5B-#8U-#5BUU
 * <86> = #6B#8U-#6BU#6B-#8U-#6BUU
 * <87> = #7B#8U-#7BU#7B-#8U-#7BUU
 * <88> = #8B#8U-#8BU#8B-#8U-#8BUU
 * <89> = #9B#8U-#9BU#9B-#8U-#9BUU
 * <8A> = #10B#8U-#10BU#10B-#8U-#10BUU
 * <8B> = #11B#8U-#11BU#11B-#8U-#11BUU
 * <8C> = #12B#8U-#12BU#12B-#8U-#12BUU
 * <8D> = #13B#8U-#13BU#13B-#8U-#13BUU
 * <8E> = #14B#8U-#14BU#14B-#8U-#14BUU
 * <8F> = #15B#8U-#15BU#15B-#8U-#15BUU
 * <92> = #2B#9U-#2BU#2B-#9U-#2BUU
 * <93> = #3B#9U-#3BU#3B-#9U-#3BUU
 * <94> = #4B#9U-#4BU#4B-#9U-#4BUU
 * <95> = #5B#9U-#5BU#5B-#9U-#5BUU
 * <96> = #6B#9U-#6BU#6B-#9U-#6BUU
 * <97> = #7B#9U-#7BU#7B-#9U-#7BUU
 * <98> = #8B#9U-#8BU#8B-#9U-#8BUU
 * <99> = #9B#9U-#9BU#9B-#9U-#9BUU
 * <9A> = #10B#9U-#10BU#10B-#9U-#10BUU
 * <9B> = #11B#9U-#11BU#11B-#9U-#11BUU
 * <9C> = #12B#9U-#12BU#12B-#9U-#12BUU
 * <9D> = #13B#9U-#13BU#13B-#9U-#13BUU
 * <9E> = #14B#9U-#14BU#14B-#9U-#14BUU
 * <9F> = #15B#9U-#15BU#15B-#9U-#15BUU
 * <A2> = #2B#10U-#2BU#2B-#10U-#2BUU
 * <A3> = #3B#10U-#3BU#3B-#10U-#3BUU
 * <A4> = #4B#10U-#4BU#4B-#10U-#4BUU
 * <A5> = #5B#10U-#5BU#5B-#10U-#5BUU
 * <A6> = #6B#10U-#6BU#6B-#10U-#6BUU
 * <A7> = #7B#10U-#7BU#7B-#10U-#7BUU
 * <A8> = #8B#10U-#8BU#8B-#10U-#8BUU
 * <A9> = #9B#10U-#9BU#9B-#10U-#9BUU
 * <AA> = #10B#10U-#10BU#10B-#10U-#10BUU
 * <AB> = #11B#10U-#11BU#11B-#10U-#11BUU
 * <AC> = #12B#10U-#12BU#12B-#10U-#12BUU
 * <AD> = #13B#10U-#13BU#13B-#10U-#13BUU

* <AE> = #14B#10U-#14BU#14B-#10U-#14BUU
 * <AF> = #15B#10U-#15BU#15B-#10U-#15BUU
 * <B2> = #2B#11U-#2BU#2B-#11U-#2BUU
 * <B3> = #3B#11U-#3BU#3B-#11U-#3BUU
 * <B4> = #4B#11U-#4BU#4B-#11U-#4BUU
 * <B5> = #5B#11U-#5BU#5B-#11U-#5BUU
 * <B6> = #6B#11U-#6BU#6B-#11U-#6BUU
 * <B7> = #7B#11U-#7BU#7B-#11U-#7BUU
 * <B8> = #8B#11U-#8BU#8B-#11U-#8BUU
 * <B9> = #9B#11U-#9BU#9B-#11U-#9BUU
 * <BA> = #10B#11U-#10BU#10B-#11U-#10BUU
 * <BB> = #11B#11U-#11BU#11B-#11U-#11BUU
 * <BC> = #12B#11U-#12BU#12B-#11U-#12BUU
 * <BD> = #13B#11U-#13BU#13B-#11U-#13BUU
 * <BE> = #14B#11U-#14BU#14B-#11U-#14BUU
 * <BF> = #15B#11U-#15BU#15B-#11U-#15BUU
 * <C2> = #2B#12U-#2BU#2B-#12U-#2BUU
 * <C3> = #3B#12U-#3BU#3B-#12U-#3BUU
 * <C4> = #4B#12U-#4BU#4B-#12U-#4BUU
 * <C5> = #5B#12U-#5BU#5B-#12U-#5BUU
 * <C6> = #6B#12U-#6BU#6B-#12U-#6BUU
 * <C7> = #7B#12U-#7BU#7B-#12U-#7BUU
 * <C8> = #8B#12U-#8BU#8B-#12U-#8BUU
 * <C9> = #9B#12U-#9BU#9B-#12U-#9BUU
 * <CA> = #10B#12U-#10BU#10B-#12U-#10BUU
 * <CB> = #11B#12U-#11BU#11B-#12U-#11BUU
 * <CC> = #12B#12U-#12BU#12B-#12U-#12BUU
 * <CD> = #13B#12U-#13BU#13B-#12U-#13BUU
 * <CE> = #14B#12U-#14BU#14B-#12U-#14BUU
 * <CF> = #15B#12U-#15BU#15B-#12U-#15BUU
 * <D2> = #2B#13U-#2BU#2B-#13U-#2BUU
 * <D3> = #3B#13U-#3BU#3B-#13U-#3BUU
 * <D4> = #4B#13U-#4BU#4B-#13U-#4BUU
 * <D5> = #5B#13U-#5BU#5B-#13U-#5BUU
 * <D6> = #6B#13U-#6BU#6B-#13U-#6BUU
 * <D7> = #7B#13U-#7BU#7B-#13U-#7BUU
 * <D8> = #8B#13U-#8BU#8B-#13U-#8BUU
 * <D9> = #9B#13U-#9BU#9B-#13U-#9BUU
 * <DA> = #10B#13U-#10BU#10B-#13U-#10BUU
 * <DB> = #11B#13U-#11BU#11B-#13U-#11BUU
 * <DC> = #12B#13U-#12BU#12B-#13U-#12BUU
 * <DD> = #13B#13U-#13BU#13B-#13U-#13BUU
 * <DE> = #14B#13U-#14BU#14B-#13U-#14BUU
 * <DF> = #15B#13U-#15BU#15B-#13U-#15BUU
 * <E2> = #2B#14U-#2BU#2B-#14U-#2BUU
 * <E3> = #3B#14U-#3BU#3B-#14U-#3BUU
 * <E4> = #4B#14U-#4BU#4B-#14U-#4BUU
 * <E5> = #5B#14U-#5BU#5B-#14U-#5BUU
 * <E6> = #6B#14U-#6BU#6B-#14U-#6BUU
 * <E7> = #7B#14U-#7BU#7B-#14U-#7BUU
 * <E8> = #8B#14U-#8BU#8B-#14U-#8BUU
 * <E9> = #9B#14U-#9BU#9B-#14U-#9BUU
 * <EA> = #10B#14U-#10BU#10B-#14U-#10BUU
 * <EB> = #11B#14U-#11BU#11B-#14U-#11BUU

```

* <EC> = #12B#14U-#12BU#12B-#14U-#12BUU
* <ED> = #13B#14U-#13BU#13B-#14U-#13BUU
* <EE> = #14B#14U-#14BU#14B-#14U-#14BUU
* <EF> = #15B#14U-#15BU#15B-#14U-#15BUU
* <F2> = #2B#15U-#2BU#2B-#15U-#2BUU
* <F3> = #3B#15U-#3BU#3B-#15U-#3BUU
* <F4> = #4B#15U-#4BU#4B-#15U-#4BUU
* <F5> = #5B#15U-#5BU#5B-#15U-#5BUU
* <F6> = #6B#15U-#6BU#6B-#15U-#6BUU
* <F7> = #7B#15U-#7BU#7B-#15U-#7BUU
* <F8> = #8B#15U-#8BU#8B-#15U-#8BUU
* <F9> = #9B#15U-#9BU#9B-#15U-#9BUU
* <FA> = #10B#15U-#10BU#10B-#15U-#10BUU
* <FB> = #11B#15U-#11BU#11B-#15U-#11BUU
* <FC> = #12B#15U-#12BU#12B-#15U-#12BUU
* <FD> = #13B#15U-#13BU#13B-#15U-#13BUU
* <FE> = #14B#15U-#14BU#14B-#15U-#14BUU
* <FF> = #15B#15U-#15BU#15B-#15U-#15BUU

```

Auxiliary macro "K", which provides a shorthand means for rotating all but the upper layer, is likely to be applied many times when using the above macros to align centre cubies.

```
* K = WDU
```

4. First Layer Corner Location Macros

Locate first layer at the bottom and shift from top left to bottom left. Use macro "A" if the colour of the front face of the upper cubie is the same as that for the bottom face of the cube, "C" if the colour of the side face of the upper cubie is the same as that for the bottom face of the cube, and "E" if the colour of the top face of the upper cubie is the same as that for the bottom face of the cube.

```

* A = FU-F
* C = -L-UL
* E = -LULUUC

```

5. Edge Cubie Adjustment Macros

Locate final layer at top. Use "P" to exchange left side front edge set with right side top edge set such that top face relocates on left face. Use "-P" to exchange left side front edge set with top front edge set such that top face relocates on front face. To obtain matched edge sets for transfer to and from the top layer, use 2U to 15U rotations.

```
* P = FUURU-RUU-F
```

Use "Q" to invert set of edge cubies located at the left side of the front face.

```
* Q = P-UP
```

6. Final Edge Set Alignment Macros

Some adjustment of the final edge set may be required before all the final layer edge cubies achieve the same sense. Place the final edge cubie set to be aligned at the front edge of the

upper layer. Adjustment can proceed in any order until all cubies in the final edge set are aligned in the same sense.

```

<18> converts [- - - - - b b - - - - -] to [- - - - - a a - - - - -]
<17> converts [- - - - - b - - b - - - -] to [- - - - - a - - a - - - -]
<16> converts [- - - - b - - - - b - - - -] to [- - - - a - - - - a - - - -]
<15> converts [- - - b - - - - - b - - - -] to [- - - a - - - - - a - - - -]
<14> converts [- - b - - - - - - - b - - -] to [- - a - - - - - - - a - - -]
<13> converts [- b - - - - - - - - - b -] to [- a - - - - - - - - - a -]
<12> converts [b - - - - - - - - - - b] to [a - - - - - - - - - - a]

```

Note: "a" and "b" refer to the two faces of the edge cubies.
 "-" means "don't care" (ie. either "a" or "b").

```

* <12> = #2R#2RBBUU#2LUU-#2RUU#2RUUFF#2RFF-#2LBB#2R#2R
* <13> = #3R#3RBBUU#3LUU-#3RUU#3RUUFF#3RFF-#3LBB#3R#3R
* <14> = #4R#4RBBUU#4LUU-#4RUU#4RUUFF#4RFF-#4LBB#4R#4R
* <15> = #5R#5RBBUU#5LUU-#5RUU#5RUUFF#5RFF-#5LBB#5R#5R
* <16> = #6R#6RBBUU#6LUU-#6RUU#6RUUFF#6RFF-#6LBB#6R#6R
* <17> = #7R#7RBBUU#7LUU-#7RUU#7RUUFF#7RFF-#7LBB#7R#7R
* <18> = #8R#8RBBUU#8LUU-#8RUU#8RUUFF#8RFF-#8LBB#8R#8R

```

7. Single Edge Set Flip Macro for Final Layer Edge Cubies

Before proceeding further with the final layer alignment it is essential that the number of edge sets that need to be flipped is not odd (one or three). If an odd number of matched edge sets need to be flipped on the final layer (can only occur for even size cubes), use "S" to flip the set located at the front of the upper layer. Alternatively, the required result can be achieved using the "Final Edge Set Alignment Macro(s)".

```
* S = 8R8RRRBBUU8L-LUU-8RRUU8R-RUUFF8R-RFF-8LLBB8R8RRR
```

8. Final Layer Corner Location Macros

Locate final layer at top and use "T" to swap the two front corners.

```
* T = B-U-FU-B-UFUU
```

Sequences containing T, U and -U can be used to cover all other final layer corner location possibilities not covered by the use of a single T macro. Such sequences together with alternative macro definitions are as follows:

With lower right corner cubie of upper face fixed, use "UUTUTU" or "G" to provide CW movement of three remaining corner cubies.

With lower right corner cubie of upper face fixed, use "-UT-UTUU" or "-G" to provide CCW movement of three remaining corner cubies.

Use "TUT-UT" or "H" to provide diagonal swap of lower left with upper right corner cubie.

```

* G = UUTUTU
* H = TUT-UT

```

9. Final Layer Edge Cubie Location Macros

Locate final layer at top. Use "V" to obtain a CW movement, and "-V" to provide a CCW movement, of the leftmost three edge cubie sets.

* $V = LLU-FBLL-BFULL$

Sequences containing V, U and -U can be used to cover other final layer edge cubie location possibilities not covered by the use of a single V or -V macro. Such sequences together with alternative macro definitions are as follows:

Swap left with right, and up with down edge cubies (or cubie sets) using "V-UVU" or "I".

Swap left with down, and up with right cubies (or cubie sets) using "-V-U-VU" or "J".

* $I = V-UVU$

* $J = -V-U-VU$

Just one opposite pair of edge cubie sets may need to be swapped. Refer to "Final Layer Diametrically Opposite Edge Set Swap Macro" if that applies.

10. Final Layer Diametrically Opposite Edge Set Swap Macro

Locate final layer at top. Locate one of the sets to be swapped at the front and use "X" to exchange it with the edge set at the back.

* $X = 8R8RRRUU8R8RRRUU8U8UUU8R8RRR8U8UUU$

11. Final Layer Corner Twirl Macro

Locate final layer at the top. Use "Y" to provide a CW twirl of the front right corner cubie. Locate another corner cubie at the front right corner and use "-Y" to provide a CCW twirl of that cubie. Up to two applications of this macro may be required to obtain correct orientation of all corner cubies.

* $Y = -RDRFD-F$

12. Final Layer Edge Set Flip Macro

Locate final layer at the top. Use "Z" to flip edge set located at the front. Locate another top layer edge set requiring a flip at the right side and use "-Z". Up to two applications of this macro may be required to obtain correct orientation of all edge sets.

* $Z = FU-DLLUDDR$

Appendix B

Using Macros for Placement of All Centre Cubies

Because the centre cubie placement macros can be applied in any order and they don't disturb edge or corner cubie alignment, there are many ways they can be used for centre cubie placement. The following is an outline of one way of placing all centre cubies using the macros defined for the Unravel program or their equivalent algorithms for any other program implementation. The Unravel program centre cubie macros swap cubies from an upper face location to an equivalent front face location. No disturbance of any other centre cubie on the front face occurs but some rearrangement of the cubies in the same orbit on the upper face occurs (an essential outcome to conform with the even parity rule). The method described in this Appendix aligns all the centre cubies on a given face before proceeding to do the same for the next face.

The first step is to align all the centre cubies for the Down (D) face. The following steps can be used:

1. Place the nominated first face at the front (F) location (for cubes of odd size the absolute centre cubie will define the face colour whereas, for even size cubes, the user can nominate any colour for the first face).
2. Transfer any upper (U) face cubies having the first face colour to the F face so that no first face coloured cubie remains on the U face.
3. Apply WFWF and repeat the previous step for the new set of centre cubies on the U face.
4. At this stage the first face colour will appear on three faces other than the one designated as the first face. Move these one at a time to the front to transfer first face centre cubies to the U face and then down to the first face placed in the front. Alternatively, the Centre Cubie Block Transfer Macro, which is available for the Java version of Unravel, can be used to swap all first face centre cubies on a front face with those on an upper face that has been cleared of first face centre cubies.
5. Use WL to place the first face with fully aligned centre cubies at the bottom (D) location.

Edge cubies for the D face should be aligned before proceeding to align further centre cubies. Corner cubies for the D face can be aligned now or later. The alignment of centre cubies for the remaining five faces can be done before or after the alignment of the remaining edge and corner cubies. In either case the steps below are the same.

The centre cubies for the five remaining faces are progressively aligned. Perform the following steps:

1. Move any unaligned middle face other than the one to be aligned to the front.
2. Transfer all centre cubies of the colour being aligned from the F face to the U face. If the Java version of Unravel is in use, the Centre Cubie Block Transfer Macro will simplify this step.
3. Move the face to be aligned to the front.
4. Transfer all centre cubies of the required colour from the U face to the face being aligned on the front.
5. Repeat steps 1 to 4 until the chosen face is completely aligned.
6. Repeat steps 1 to 5 until only one middle face and the U face needs to be aligned.

7. Place the last of the four middle faces to be aligned at the front and transfer the required centre cubies from the U face to the F face.

Alignment of the centre cubies on the fourth middle face means the centre cubies on the fifth (U) face is also completely aligned.

Appendix C

Laws of the Size n Cube

A cube is solvable if the set state has existed some time in the past and if no tampering of the cube (e.g. by rearrangement of stickers on hardware cubes) has occurred. Laws that limit what arrangements are possible for the size 3 cube^{[10][11]} and for cubes of all sizes^{[9][12]} have been published. Those laws mean that, of the possible unrestricted cubie arrangements, the number that are unreachable far outnumber those that are reachable.

Cubes of all sizes have three mutually perpendicular axes about which one or more layers can be rotated. All moves for the cube can be considered as a sequence of quarter-turn rotations about these axes. The movement possibilities give rise to a set of laws which in most cases can be expressed in analytical terms.

Every cube move can be considered as a permutation. The relationship between the cube state after the move with that before the move can be expressed mathematically using Group Theory^{[3][4][5]} to quantify permutations. Since every move can be considered as a sequence of quarter turn rotations, it is appropriate to examine what is involved in a quarter turn rotation. Except for the absolute centre cubie for odd size cubes, during the quarter turn the cubies move in separate four-cubicle trajectories (also referred to as a 4-cycle movement since four quarter turns will restore the cubies in the specified trajectory to their original positions). A quarter turn of a 4-cubie set can be represented by three swaps as indicated below where swap (12) means the contents of cubicle 1 is swapped with the contents of cubicle 2, etc.

Start		Swap (12)		Swap (13)		Swap (14)	
1/A	2/B	B	A	C	A	D	A
4/D	3/C	D	C	D	B	C	B

The parity^[6] of a permutation refers to whether that permutation is even or odd. An even permutation is one that can be represented by an even number of swaps while an odd permutation is one that can be represented by an odd number of swaps. An odd permutation followed by an odd permutation will represent an overall even permutation (adding two odd numbers always returns an even number). Since a quarter turn is made up of a number of 4-cycles each involving three swaps, if the number of 4-cycles is odd, overall parity of the quarter turn permutation will be odd and vice versa.

Quarter turn permutation parity for a size n cube is given in the following table.

Cube size (odd or even)	Layer type	Number of 4-cycle movements	Overall parity
odd	inner	$n - 1$	even
	outer	$((n - 2)^2 - 1)/4 + (n - 1)$	even*
even	inner	$n - 1$	odd
	outer	$((n - 2)^2)/4 + (n - 1)$	even if $n/2$ is even ** odd if $n/2$ is odd

* Since $((n - 2)^2 - 1)$ equals $(n - 1)(n - 3)$, a product of two consecutive even numbers, which must always be evenly divisible by 8.

** Since $((n - 2)^2)/4$ equals $(n/2 - 1)^2$, and will be odd if $(n/2 - 1)$ is odd (i.e. if $n/2$ is even) giving overall even since $(n - 1)$ is odd. The reverse applies if $n/2$ is odd.

Summarising the above parity results we conclude:

- All permutations for odd size cubes have even overall parity.
- All individual quarter turns for even size cubes, where half the cube size is an odd number, have odd overall parity.
- For even size cubes where half the cube size is an even number, inner layer quarter turns have odd overall parity and outer layer quarter turns have even overall parity.

The above analysis considered the parity for corner (where applicable), edge and centre cubies combined. It is possible to consider these in isolation and when that is done an even combined quarter turn parity will involve a number of odd parity elements.

Cubes of any size greater than 3 behave exactly like the size 3 cube if only outer layer rotations are permitted. Parity rules dictate that a single pair of edge sets cannot be swapped using outer layer rotations only. Swapping two pairs maintains even parity and is possible.

A permutation as used here takes account of the change in the positions of cubies, but not any change in their orientations. Each outer layer move always flips an even number of edge sets, so it is not possible to flip an odd number of edge sets using only outer layer rotations.

Corner cubies behave the same for cubes of all sizes. They have three possible orientations made up from a combination of $1/3$ twists where a full twist (about an axis drawn from the cube corner to the cubie's internal corner) returns the corner cubie to its original orientation. If we designate a unit clockwise twist by $1/3$ and a unit counter-clockwise twist by $-1/3$, then the twist possibilities for a corner cubie relative to any given initial state (the set state for example) are 0, $1/3$ and $-1/3$. The sum of the twist increments across all corner cubies must always be an integer (0, 1 or 2).

When inner layer rotations are included for cubes of size greater than 3, some of the edge cubie movement limitations mentioned above no longer apply. These are expanded in *Appendix D: Final Layer Problems for Cubes of Size 4 and Above*

Cubie position and orientation are of special concern when unscrambling the final layer. Edge cubies must always end up in exactly the same positions they occupied in the initial set state before scrambling. If any edge cubie in a given edge set in the final layer has a wrong orientation (only applicable to cubes of size greater than 3), it must be in a wrong position and will need to be swapped with a complementary edge cubie also having wrong orientation. With everything else in place, corner cubies can be in the right position but two or more can have incorrect orientation. For cubes of size greater than 3, centre cubies are extremely unlikely to occupy the same positions they had in the initial set state (assuming the centre cubies are unmarked).

If permutations of facelets rather than cubies are considered then both position and orientation of cubies will be taken into account. For software cubes, the states (six color possibilities) of the $6n^2$ facelets (in a $6 \times n \times n$ array for instance) is what would allow complete information on cube state to be saved for later use.

A cube of any size that is subject to repeats of the same permutation will eventually return to the state (e.g. the set state) it occupied before the first application of the permutation^{[3][4][9]}. The number of times that a permutation has to be applied to return the cube to its initial state is referred to as the order or the cycle length of the permutation.

Appendix D

Final Layer Problems for Cubes of Size 4 and Above

A final layer problem is defined here to mean a need for a rearrangement of cubies that cannot be achieved using standard size 3 cube moves. The moves (algorithms or macros) on which the graphics in this section are based on are the “Final Edge Set Alignment Macro(s)”, the “Single Edge Set Flip Macro for Final Layer Edge Cubies” and the “Final Layer Diametrically Opposite Edge Set Swap Macro” that are detailed in “Sec. 10: Final Layer Actions” in the main document.

For cubes of size greater than 3, edge cubies can occupy positions and/or orientations that are at variance with what can arise for the size 3 cube. These are often referred to as parity problems or errors, but such terminology may be misleading. If moves were limited to those available to the size 3 cube, such states would be unreachable (break parity rules). The parity laws, as defined in *Appendix C: Laws of the Size n Cube*, must be obeyed in all permutations and the corrections referred to here are no exceptions. There are numerous variations in the manner of presentation and the algorithms to resolve the final layer problems, but the correction requirement will be similar to those described below. The problems for larger cubes can be considered as straight forward extensions of those that apply to the size 4 cube. While there are many ways of rearranging the cubies to overcome final layer problems, all cubes can be solved with attention to two basic problem situations:

- There is a need to flip a complementary pair or a complete set of edge cubies in a final edge set. This condition will be referred to as an OLL (orientation of last layer) requirement.
- There is a need to swap the positions of two edge cubie sets in the final layer. This condition will be referred to as a PLL (permutation of last layer) requirement.

OLL⁷ and PLL⁸ as used here can be considered a sub-sets of usual definitions^[13] of these terms.

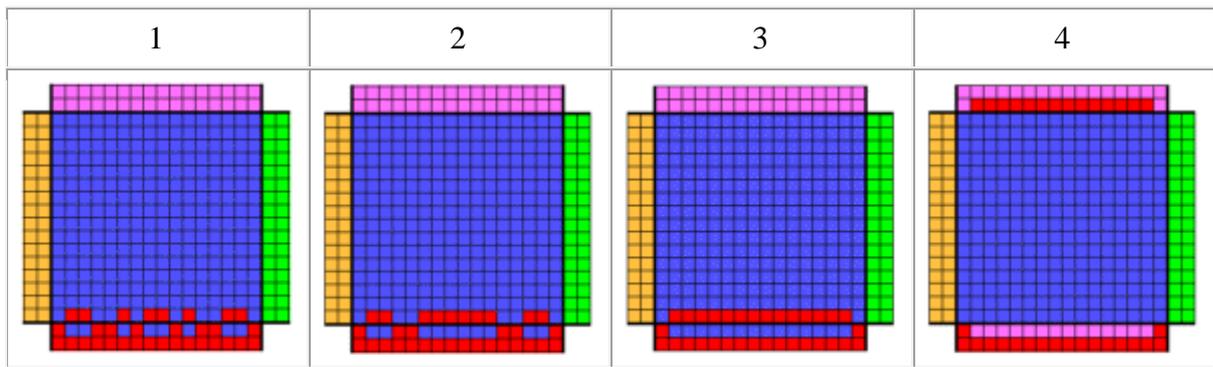
The problems can be avoided if both edge and corner cubies are put in place before the centre cubies. However, avoidance of these problems should not be considered to be very significant for cubes of large size since the corrections are well defined and easy to apply. Furthermore, even when edge and corner cubies are aligned first, there is still a need to involve centre cubie adjustments to achieve that. Basically, a problem exists in the form presented here if:

- all centre cubies appear to be correctly placed, and
- eleven of the twelve edge cubie sets have been aligned, and
- the sense of the cubies in the 12th edge set is not all the same, or
- the sense of the cubies in each final layer edge set is the same but the positions of two of the edge sets need to be swapped.

The problems that can arise for a size 16 cube example are illustrated below where departures from a completely solved cube are shown. In practice these problems are likely to be corrected before other alignment actions are completed for the final layer.

⁷ OLL usually means manipulating the last layer cubies so that the face has uniform colour, even at the expense of incorrect colours on other sides.

⁸ PLL usually means moving the last layer cubies to correct positions while preserving their orientation.

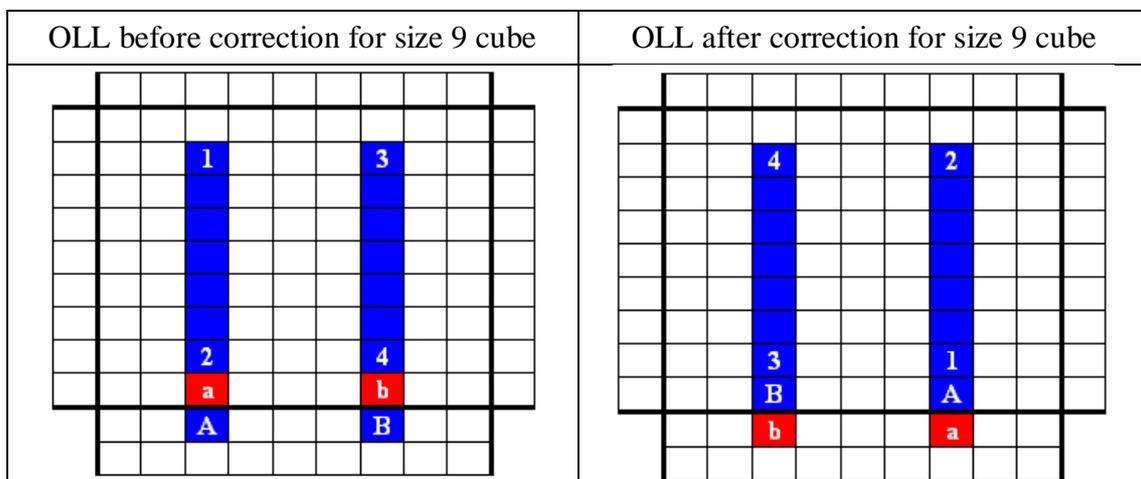


1. Final edge set in this example has three complementary pairs of edge cubies that have a different sense to the other edge cubies in the set.
2. An appropriate sequence of rotations is applied to 1 to correct the problem for one of the complementary pairs (similar sequences are required to perform corrections on the remaining complementary pairs still requiring correction).
3. The sense of all the edge cubie elements in the final set is the same but just this one set needs to be flipped to resolve this problem (this can only occur for cubes of even size). Alternatively, action as indicated here can be avoided by flipping initially the inner complementary pair of edge cubies to match the correct final alignment and then matching all further complementary pairs of edge cubies to the initial pair.
4. The positions of two (and only two) edge cubie sets need to be swapped to resolve this problem (can occur only for cubes of even size). The solution to this problem can also be subdivided into a sequence of moves^[9] that applies to just four edge cubies at a time (as for the size 4 cube).

Items 1 to 3 refer to the OLL problem resolution and item 4 refers to the PLL problem resolution^[14].

The correction of all of the problems mentioned above involves some re-arrangement of centre cubies. Problems similar to those illustrated in images 1 and 2 occur in odd size cubes. Rotation sequences required to resolve final layer problems for cubes of any size are similar to those required for the size 4 cube.

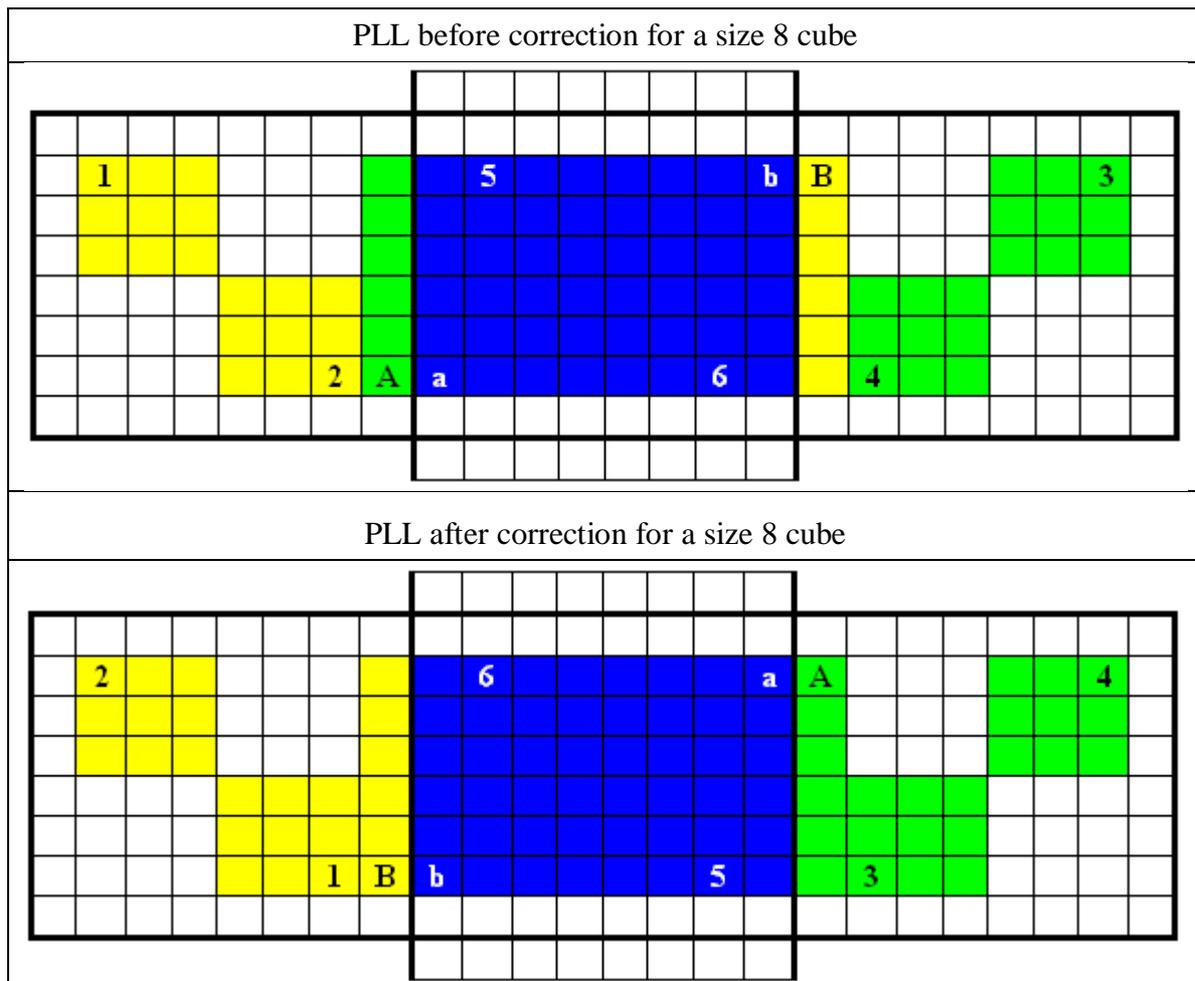
Because there are no distinguishing marks on the centre cubies when OLL and PLL corrections are made, it can be difficult to visualize what is happening. In the following illustrations the only cubies (centre and edge) that move position are shown in colour. Alphanumerical identification shows positional movement.



For the OLL correction there are $(n - 2)$ centre cubie swaps and overall there are $(n - 1)$ swaps when the edge pair is included. For odd size cubes $(n - 1)$ is always even (and conforms to the universal parity requirement for odd size cubes). For even size cubes $(n - 1)$ is always odd which means in this case a parity reversal always occurs, an allowable parity condition for even size cubes.

For the complete edge set flip (a requirement that can arise only for cubes of even size), there will be $(n - 1)(n/2 - 1)$ swaps. The overall number of swaps will be even if $(n/2 - 1)$ is even (i.e. $n/2$ is odd). The overall number of swaps will be odd if $n/2$ is even.

A typical PLL correction for a size 8 cube is illustrated below.



For this PLL correction there are $(n - 2)^2$ centre cubie swaps and since this correction is required only for even size cubes, that value is always even. Likewise the number of edge cubie swaps $(n - 2)$ is also even. Hence overall even parity always applies for this permutation.

Initially consider the number of edge cubie flips required for a size n cube. If the centre pair for even size cubes or the middle edge cubie for odd size cubes, is left unchanged and all complementary pairs that don't match are flipped, then the maximum number of corrections (including the complete edge set flip as in image 3) that would be required for an even size cube is $(n - 2)/2$, and that for an odd size cube is $(n - 3)/2$. For large cubes, on average, one would expect about half the maximum number of swaps would be required (much like tossing a coin many times and observing that the heads count tends towards being the same as the tails count). For large cubes, one would expect that approximately $n/4$ (a relatively small number) edge cubie flip corrections would be required.

The PLL correction illustrated in image 4 is either not required or to be applied once for cubes of any even size greater than 3. Swapping any two edge sets (either opposite or adjacent) will resolve the edge-cubie-set position problem. Any further adjustments required to the final layer can be achieved using standard final layer algorithms for use with the size 3 cube.