Graph Application in Rubik's Cube for Blindfolded Solving

Dhafin Rayhan Ahmad 13518063[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]*13518063@std.stei.itb.ac.id*

***Abstract*—For years, people have been developing their knowledges and skills to explore more about the twisty puzzle Rubik's cube. Various modification and customization have been applied to the original woody block hardware. The solving methods are also growing to push down any limitation exist in the world of speedsolving. When blindfolded solving were introduced to the community, the tricky mechanism of the Rubik's cube is getting more covered in studies about the cube, including the possible relation between mathematics graph and the cube itself.**

***Keywords*—Rubik's cube, graph, blindsolving, tracking**

I. INTRODUCTION

Over the decades, people have been spending their time just on a cubic twisty puzzle, the Rubik's cube. Since its popularity in 1980s, a lot of puzzle-enthusiast had tried to beat each other in term of solving time. Some even tried to solve the cube in various way, such as one-handed, with feet, or the hardcore one, blindfolded. Several methods have been invented in order to solve the Rubik's cube. Most of the methods combine the advantages of intuition and memorization. On the other hand, the blindfolded solving requires a one to have strong intuition about how the pieces are moving in the Rubik's cube. Therefore, people with great intelligence – notably high IQ point – will most likely to learn blindfolded solving faster than the others. Although, with a lot of practice, anyone can achieve the same ability.

With eyes closed while solving the Rubik's cube, one can not track on where the pieces are going. So before starting the solve, he should memorize where the pieces should go (with eyes open, of course), and these information are used to determine his turns while on the solving stage. This blindfolded phase requires him to turn the cube very carefully, as a single mistake can already ruin the whole solve. But with a lot of practice, and an appropriate turning technics – also called "fingertricks" – people can smash the blindfolded phase just as fast as a normal solve, without worrying about messing it up. Although, no blindfolded solver can guarantee a one hundred percent of success on that speedy rhythm.

Blindfolded solving – or the abbreviation, blindsolving – methods are keep being developed in years. Some blindsolvers are inventing the easier method on solving the cube blindfolded, which aim to approach more people into blindsolving. While the others, are keep inventing a more advanced one, as an attempt to push their limits down with optimal memorization and execution methods. But what we have to know is that, all these methods are based on a basic concept, often called "piece-tracking". This concept can even be made much simpler with the present of graph representation. With a good understanding on the relation between the two, one can already dig into the world of blindfolded solving.

II. RUBIK'S CUBE, BLINDSOLVING, AND GRAPH

While there are several applications of graph theorem in this paper, all of those theorems will not use any complicated mathematics result and will depend on the basic graphing and relations that we all are familiar with.

A. Rubik's Cube

The Rubik's cube is a six-sided three-dimensional puzzle, each side usually colored with different colors each other. The cube was invented in 1974 by Hungarian sculptor and professor of architecture Ernő Rubik, and get licensed to be sold by Ideal Toy Corporation in 1980. After this agreement, the puzzle start to spread wide around the world, starting the crazy age of solving the Rubik's cube.

The sides of Rubik's cube are recognized with the difference of sticker color sticked on the cube surface, each one of these six colours: white, yellow, green, blue, red, and orange. A popular coloring scheme is to place two similar colors at the opposite side to each other (e.g. red is opposite to orange), although other forms of scheme are also found in the other part of the world, such as the Japanese-scheme in Japan. Some people even use different colors on the cube, using their own color choice in contrast of the six popular one.

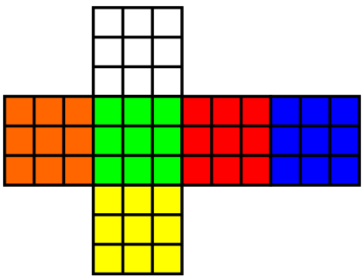

Fig. 1. The standard color scheme for the Rubik's Cube.

The Rubik's cube can be seen as a set of faces, which each side has 9 stickers, resulting in a total of 54 faces. A better way to view the cube is to partition the cube into pieces, which is more practical in any solving method. This also gives anyone who learns the cube a better understanding of how the mechanisms work. The puzzle is broken down into three main parts, which is the corners, the edges, and centers. On a normal 3x3x3 Rubik's cube, there are 8 corner pieces, 12 edge pieces, and 6 center pieces. A corner piece can be defined as the piece that has three colors on it, edge piece is where the piece has two colors on it, whereas the center piece has only one color attached to it.

Fig. 2. From left to right, showing the Rubik's cube in highlight of: the corner pieces, the edge pieces, and the center pieces.

A corner piece can only go to the place of the other corners, and so for edges and centers. By this means that, for example, an edge piece can not go to the place of a corner piece. The position of center pieces are not changing to each other by any turn, so they are already in their fixed position. Due to this fact, any turns applied to the cube are just actually edges and corners messing around the center pieces.

Some terms are defined for the Rubik's cube to make discussion easier. In a blindfolded solving world, it is required to be familiar to the terms of Rubik's cube turning, memorizing, and tracking. Notations are used to make the terms even simpler to understand. These terms will be discussed in a later section.

B. Blindsolving

Blindsolving, or blindfolded solving, is used to describe the action where someone is attempting to solve the Rubik's cube blindfolded, using the information he got before by observing the cube with eyes open.

Cubers – a term for people who like to play the Rubik's cube and it's variant – have developed several methods in solving the Rubik's cube blindfolded. The method of blindfolded solving is different with sighted solving. While on a sighted solve you can see what case you're getting on after applying some moves, in blindfolded solve you have to know where the pieces are going mid-solve.

Fig. 3. A cuber attempting a blindfolded solve – unaware with the situation of his house.
(Source: https://ruwix.com/the-rubiks-cube)

The base concept of doing a sighted solve is to complete the cube layer by layer, and the more advanced methods might do the layers simultaneously. On the other hand, blindsolving has the base concept called "tracking." During the memorization phase, one who do a blindfolded attempt will try to track on two stuffs: where the corners are going, and where the edges are going; each of them started with a piece of his choice as a "buffer." These tracking are translated into the memorization method he prefers, usually by using a letter representation for each sticker, and forms words from them. This memorization later be translated on the solving phase, or the execution phase, by doing algorithms that only affect few pieces once, so it is easier to maintain the tracking of the whole cube.

C. Graph

A graph $G = (V, E)$ is a discrete structure consists of a set of objects $V = \{v_1, v_2, \ldots\}$ called vertices (or nodes), and another set $E = \{e_1, e_2, \ldots\}$ in which the elements are called edges, such that each edge e_k is identified with an unordered pair (v_i, v_j) of vertices.

Graphs are usually visualised by a diagram, where the vertices are represented as points and each edge as a line segment from one point to another. These kind of visual diagram representation are often referred as the graph itself.

The graph has terminology for easier discussion. Some of the importants are mentioned here.

1. Adjacent

Two vertices are to be called adjacent if both of them are connected directly by an edge. Or formally, for a node v_i and node v_j there exist an edge e such that $e = (v_i, v_j)$.

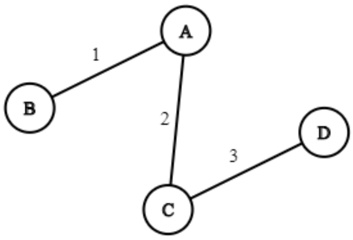

Fig. 4. A graph of four nodes. Node A is adjacent with node C, but not with node D.

2. Incidency

If an edge have one of its end connected to a node, we call it incidency. For any edge $e = (v_i, v_j)$, e is having incidency with the nodes v_i and v_j. In Fig. 4, the Node C is having incidency with edge 2 and edge 3.

3. Isolated vertex

An isolated vertex is a node that does not have any incidency edge within it. Fig. 5 gives an idea about the definition.

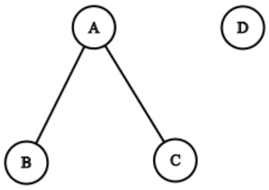

Fig. 5. A graph with an isolated vertex D.

4. Null graph

A graph is said to be a *null graph* it does not have any edges.

5. Degree

Degree of a vertex is the number of incidency edges it has. For example, the degree of Node A is two in Fig. 1, whereas the degree of node C is one. Vertex D is an isolated vertex and therefore it has a degree of zero.

6. Path

A path of length n from a starting vertex v_0 to destination vertex v_n in a graph G is an interchanging set of $v_0, e_1, v_1, e_2, \dots, v_{n-1}, e_n, v_n$ so that $e_1 = (v_0, v_1), e_2 = (v_1, v_2), \dots, e_n = (v_{n-1}, v_n)$ are the edges of graph G. In Fig. 6, the vertices A, C, D create a path.

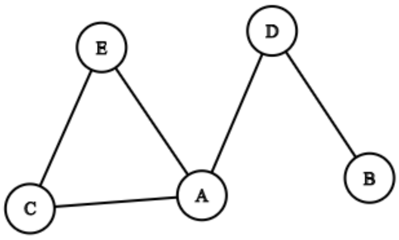

Fig. 6. A graph of five nodes.

7. Connected graph

A graph is connected if any vertex on it has at least a path to every other vertices. A graph is unconnected if there exist two vertices that does not have a path connecting them.

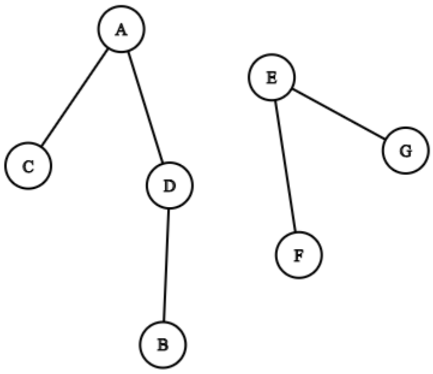

Fig. 7. A disconnected graph.

III. Notation and Convention

Several notations are used to cover the discussion of turns, permutations, and piece names in the Rubik's cube. Some conventions are also to be introduced to keep the explanation simple.

The Rubik's cube would generally be drawn as a three-dimensional cube showing three sides of it, while the rest are not shown due to perspective angle. Some figures will use translucent cube drawing to show the back sides of the cube. The six sides of the cube are named by the direction their faces are pointing to (*up*, *down*, *left*, *right*, *front*, *back*). The convention is to say the side that appears on the upper part of the figure is *up*, the one on the left is *front*, and the other one is *right*. Therefore, the sides not shown on the figure are *down*, *back*, and *left*, each of them are opposite to *up*, *front*, and *right*, respectively. In the default solved state case shown on this paper, as appears on Fig. 8, the *up* would be the yellow side, *front* would be the red side, and the green side for *right*.

Fig. 8. The three sides of the cube, each letter denoting the initial of their side names.

From here on, we will be using a shorter way to call the six sides, by their initials. So for example U is for *up*, R is for *right*, and so on. On a turning sequences, or called an algorithm, moves are written as the six initials, telling which face needs to be turned 90° clockwise. An apostrophe modifier tells us to turn the side 90° counterclockwise (instead of clockwise). Another modifier is to put the number 2 at the end of a letter to denote the 180° turn. Algorithms should be executed in the order as they appear. An example algorithm is $R\ U'F2$, which read as "turn the *right* face 90° clockwise, and then turn the *up* face 90° counterclockwise, and then turn the *front* side 180°."

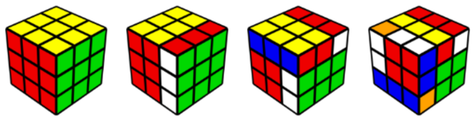

Fig. 9. From left to right: the puzzle on its solved state; R applied; $R\ U'$ applied; and finally the whole algorithm $R\ U'F2$ is applied.

Beside face turns, there are also turns that affect the middle part of the cube called slice turns. The three slice turns are M, S, and E. Slice M is the slice between *left* and *right* sides, being turned just as the L move. Slice S is the one between *front* and *back* sides, following the turning of F move. The last, E slice is between up and down sides, turns as the D face turns. Modifiers also applies to slice moves. Fig. 10 shows the solved state cube after being applied by the slice moves.

Fig. 10. Appereance of the cube after being applied: the M move, the S move, and the E move; respectively, each from a solved state.

Another important thing to note is the commutators and conjugates notation. A commutator is an algorithm in the form of $A\ B\ A'\ B'$, where either A and B can be a set of turns or just a single turn. Whenever an algorithm meet this condition, it can be written in a shorter notation, $[A, B]$. For example, the commutator $[U'R2\ B, L]$ in it's longer form is $U'\ R2\ B\ L\ B'R2\ U\ L'$. Notice that the inverse of an algorithm is made by reading the algorithm backward and inversing every individual move on it (180° turn moves mantain the same).

A conjugate is where an algorithm is in the form of $A\ B\ A'$. The form can be written as $[A: B]$. We can combine conjugates and commutators on commutator, as in $[D: [U'\ R'\ U, M']]$, which read as $D\ U'\ R'\ U\ M'\ U'\ R\ U\ M\ D'$.

To build an easier communication, the community of Rubik's cube define a standard guide for naming each sticker place for the cube. Rather than saying "the yellow sticker on the green-yellow-red corner," it is more convenient to say it by the initials of the face associated to the sticker. For example, we would refer the red sticker on the yellow-red-green corner as FRU. The first initial is indicating on which side is the sticker we're meant to, followed by another two sides of the corner, preferably in counterclockwise cycle (saying FUR is still acceptable though). The same goes for the edges, for instance, the yellow sticker on the green-yellow edge is called UR. Please be aware that this labelling system is dependent on the cube orientation we use.

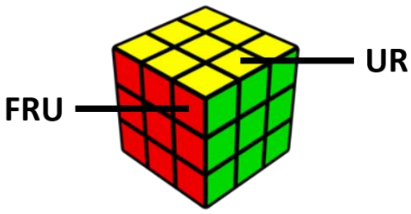

Fig. 11. Example of the labelling system.

Another easy way of naming the stickers is by using letters. For blindfolded solving, it is often encouraged to use this lettering system to make the memorization easier. The letters translate from the normal labelling system as shown on Table 1.

TABLE I. THE LETTERING SYSTEM FOR CORNERS

Sticker Label	Sticker Letter
UBL	A
URB	B
UFR	C
ULF	D
FUL	E
FRU	F
FDR	G
FLD	H
RUF	I
RBU	J
RDB	K
RFD	L
BUR	M
BLU	N
BDL	O
BRD	P
LUB	Q
LFU	R
LDF	S
LBD	T
DFL	U
DRF	V
DBR	W
DLB	X

As seen on Table 1, capital letters are used. To make it different, the edge stickers use noncapital letters instead.

TABLE II. THE LETTERING SYSTEM FOR EDGES

Sticker Label	Sticker Letter
UB	a
UR	b
UF	c
UL	d
FU	e
FR	f
FD	g
FL	h
RU	i
RB	j
RD	k
RF	l
BU	m
BL	n
BD	o
BR	p
LU	q
LF	r
LD	s
LB	t
DF	u
DR	v
DB	w
DL	x

IV. Graph for Representing Piece Cycles

As we learned before, a corner sticker can only go to another corner sticker, and so do for edges. But notice that not every position can be reached by a corner with a single move. For example, the H sticker can go the the position of L sticker with just D move, but requires a minimal of two moves to get the position of C sticker. If we create a graph with its vertices being the edge stickers and corner stickers, the edges being connection between stickers to other stickers that can be reached in a single move, we can apply some interesting graph algorithms there. For example, we can use Breadth First Search algorithm to find the shortest path from a sticker to another, which really help in setting up pieces while doing a blindsolve.

Fig. 12 shows the graph we have mentioned before. This graph, call it G, has the vertices $V = \{A, B, \dots, X, a, b, \dots, x\}$ and $E = \{(A, B), (A, C), \dots\}$. Since edge stickers can not go into the position of a corner sticker, and vice verca, therefore the graph G is a *disconnected graph*.

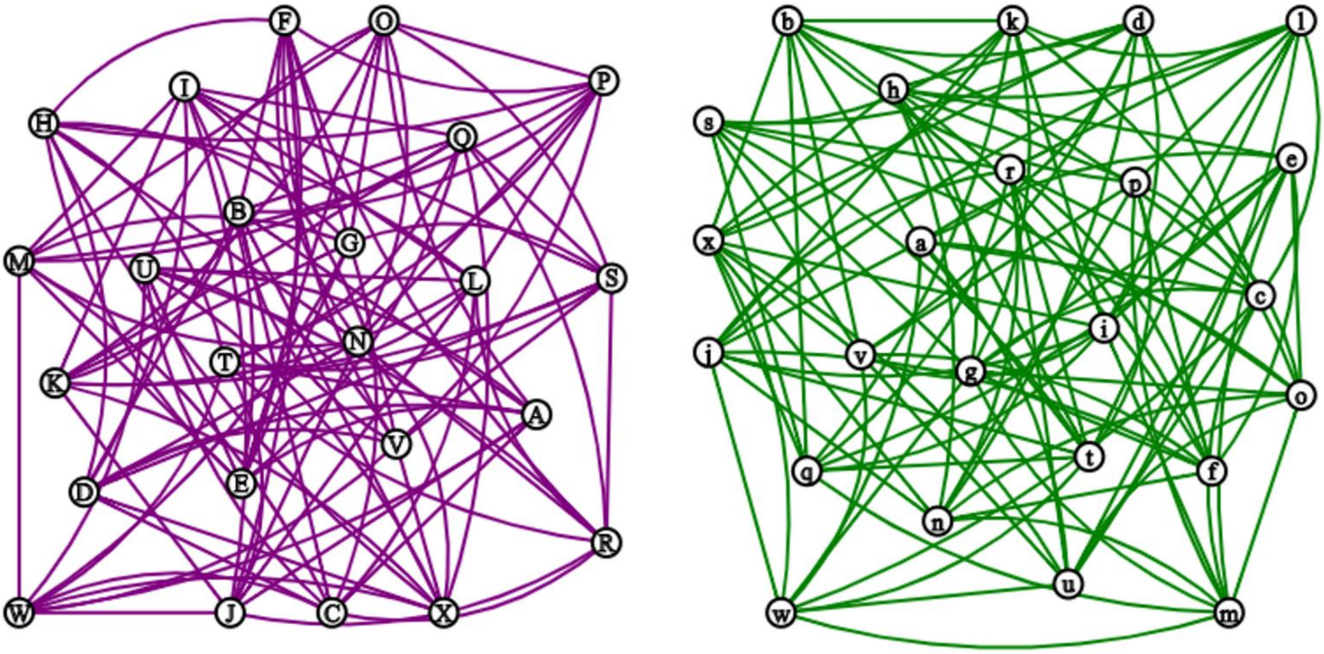

Fig. 12. The disconnected graph showing connection between pieces that can be reached by a single turn.

An example use of this graph is the piece set-up part on the beginner method of blindsolving, *OP/OP*. This execution method was founded by Stefan Pochmann, before he invented his new method for corners which he called R2, and a method for edges named M2. OP is the abbreviation for *Old Pochmann*, called this way as it's Stefan's older method. OP/OP means doing both edges and corners execution with his old methods.

The concept is the method is simple, it use buffer pieces as a starting piece, and then we track on where the piece should go to get the solved state. The buffer piece for corners is UBL, and the buffer for edges is UR. An algorithm is used to shoot the corner buffer to a target sticker, which is on the position of RFD. A different algorithm is used to shoot the edge buffer to a target sticker on UL. To shoot the buffer to a different target mentioned before, we use set-up moves.

To determine the set-up moves, we first eliminate the side that should not be turned on the set-up. For corners, since the buffer is on UBL, therefore the sides U, B, and L can not be a part of the set-up algorithm. This means that our set-up algorithm would contains only combination of D, F, and R turns. This situation can be drawn on a new graph (showing only corners relation) as on Fig. 13.

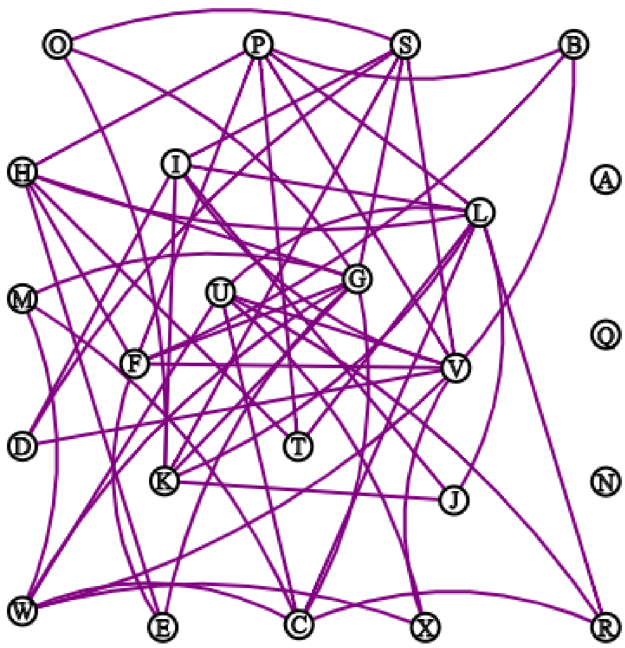

Fig. 13. Graph of corner stickers relation in OP/OP blindsolving method.

The connection graph of corner stickers is simpler than the original graph. This is due to the additional limitation to the turns. Notice how the sticker A, Q, and N are isolated vertices now since they are the stickers on the buffer piece.

The graph can also be represented as an adjacency matrix as shown in Fig. 14 below.

$$
\begin{matrix}
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&1&0&0\\
0&0&0&0&0&0&1&0&0&0&0&1&1&0&0&0&0&1&0&0&1&0&1&0\\
0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&1&0&0&1&0&0\\
0&0&0&0&0&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&1&0&1&1&0&0&0&0&0&0&0&1&0&0&0&0&0&1&0&0\\
0&0&1&0&1&1&0&1&0&0&1&0&1&0&1&0&0&0&1&0&0&0&1&0\\
0&0&0&0&1&1&1&0&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&0\\
0&0&0&1&0&0&0&0&0&1&1&1&0&0&0&0&0&0&1&0&0&1&0&0\\
0&0&0&0&0&0&0&0&1&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&1&1&0&1&0&0&1&0&0&0&1&0&0&0&0&0\\
0&0&1&0&0&0&0&1&1&1&1&0&0&0&0&1&0&1&0&1&1&0&0&0\\
0&0&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0\\
0&1&0&0&0&1&0&1&0&0&0&1&0&0&0&0&0&0&0&1&0&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&1&0&0&0\\
0&0&0&1&0&0&1&0&1&0&1&0&0&0&1&0&0&0&0&0&0&1&0&0\\
0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&1&1&1\\
0&1&0&1&0&1&0&0&1&0&0&0&0&0&0&1&0&0&1&0&1&0&1&1\\
0&0&1&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&1&1&0&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&0
\end{matrix}
$$

Fig. 14. The $24x24$ adjacency matrix for showing the relation of corner movements. Both row and column are labelled alphabetical starting from A to X.

For edges, it follows the same rules as in corners, where the illegal moves for setting-up pieces are: U, S, and R. This also results in a simpler graph for the connection representation of the edges.

A more advanced method uses direct 3-cycle to move the pieces along the solves. The method is often referred as *3-style*. This idea of this method is to use the commutator concept to move three pieces at a time without messing up the others. An example commutator would be $[R\ U\ R', D]$ which move UFR to RFD, RFD to FLD, FLD to UFR. It forms a 3-cycle of $UFR - RFD - FLD$. A more advanced application of matrices can be applied onto this method discussion.

V. Appendix

All Rubik's cube models shown in this paper are generated from an open source visual cube generator from http://cube.crider.co.uk/visualcube.php. All graph model visualization are generated from CSAcademy's graph generator at https://csacademy.com/app/graph_editor.

VI. Acknowledgment

All praise to Allah, only with His guidance I could finish this paper. After that, I thanks Fariska Zakhralativa Ruskanda, S.T., M.T. for her guide in understanding the discrete mathematics in the class. Special thanks to Ernő Rubik for his amazing invention on the cube. I also appreciate to the great community of cubing, especially Stefan Pochmann, for developing such a great concept in the world of blindfolded Rubik's cube solving.

References

[1] K. H. Rosen, *Discrete Mathematics and Its Applications Seventh Edition.* New York: McGraw-Hill, 2012.

[2] N. Deo, *Graph Theory with Applications to Engineering and Computer Science.* Englewood Cliffs, NJ: Prentice-Hall, 1974.

[3] R. Munir, *Diktat Kuliah Matematika Diskrit*. Bandung: Program Studi Teknik Informatika STEI, 2003.

[4] https://www.rubiks.com/en-us/about (*accessed on December 5th, 2019*)

Pernyataan

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 6 Desember 2019

Dhafin Rayhan Ahmad (13518063)