A review of mathematical methods for construction of Gray codes

Urvashi Arora

Department of Mathematics
Rajdhani College, University of Delhi,
New Delhi-110015,INDIA.
urvashi.arora@rajdhani.du.ac.in

Abstract

Gray codes which are an ordered collection of sequences of bits zeros and ones satisfying certain specific conditions are used extensively in digital electronics for minimizing the amount of switching and for improving reliability of switching systems. They are used in applications in which an ordinary sequence of binary digits produced by the hardware may cause an error or create an ambiguity while transitioning from one number to the next number. Gray codes are used to simplify the process of error correction in digital communication systems. Gray codes are also used for constructing switching circuits. Construction of Gray codes is a problem of concern while designing switching circuits to improve dependability of the circuits. The problem of construction of Gray codes using mathematical methods is reviewed in this paper. Construction of Gray codes using the direct method via Hamiltonian graphs as well as an iterative method that can be computationally implemented have been analyzed and compared. The real life applications of Gray codes have also been briefly discussed.

Subject Classification:[2010]Primary 05C45, 05C85; Secondary 68R10, 94Axx
Keywords: Gray, codes, discrete, graph, Hamiltonian.

1 Introduction

In the field of discrete mathematics, a branch of study of immense importance is that of graph theory. Graph theory basically consists of diagrams which are called ‘graphs’. In simple mathematical language, a graph may be defined as a diagram consisting of points (which are called ‘vertices’ mathematically) and lines (which are called ‘edges’ mathematically). Graph theory finds numerous applications in various fields such as computer science, linguistics, physics, chemistry, computer networks, social sciences and so on. In the field of electrical engineering, graph theory is used for designing circuits, where the circuit connections are termed as topologies. There are several topologies such as series topology, star topology, parallel topology etc. An example of a graph G with the set of vertices $V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$ and set of edges $E = \{e_1, e_2, e_3, e_4, e_5, e_6\}$ is as follows as in figure1.

There is a special category of graphs called ‘Hamiltonian graphs’ named after Sir William Rowan Hamilton who came out with a game in 1857 which involved searching

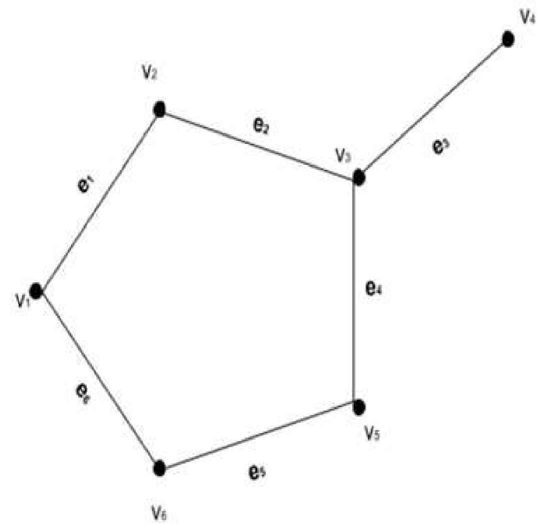

Fig. 1: **An example of a graph**

for a Hamiltonian cycle in a graph [1]. Hamiltonian cycles have various applications such as scheduling of time, choosing travel routes, choosing of network topologies etc [2].

One important application of Hamiltonian cycles is their use in construction of Gray codes (invented by Frank Gray). Alternatively called Reflected Binary Code (RBC), a Gray code, is an arrangement of the binary number system sequences in which two consecutive sequences vary from each other in exactly one binary digit 0 or 1 with the first and last sequences also differing by one bit. As an example, the decimal number '1' and '2' may be represented as 001 and 010 respectively in the binary system. But using a Gray code '1' and '2' are represented as 010 and 011 respectively. This means that for increasing the value of a variable from decimal number '1' to decimal number '2' using a Gray code requires changing of only one bit instead of two bits as in the case when decimal numbers are represented using a binary system/code. This fact makes Gray codes very popular in communication systems dependent on digital data for reducing the amount of switching in digital circuits. Also Gray codes help in eliminating serious consequences that can occur due to ambiguity arising in applications dependent on normal sequences of binary system. As an example, suppose in a system dependent on binary system of representation, if the system undergoes a change from the initial state of decimal digit '3' (i.e. 011) to the final state of decimal digit '4 '(i.e. 100).Since the binary representations of '3' and '4' differ in all the three bits, therefore a change of all the three bit positions is required to change from '3' to '4'.Suppose the change takes place through the following states 011(initial state),001,101,100(final state).The chances of a state being read wrongly during transition are high in such a case and thus it can finally lead to ambiguous results. If instead a Gray code of length 3 given by: 000, 010, 011, 111, 110, 100, 101, 001 is used, then transitioning from '3' to '4' would simply mean changing from 111 to 110 which requires a change of one bit only and therefore reduces the possibility of errors. Gray codes were used for solving several puzzles in mathematics such as Chinese rings puzzles, towers of Hanoi game, towers of Bucharest game etc, before they found use in digital electronics [3].Now a days

Gray codes are used for error correction in digital terrestrial television (DTT) broadcasting which is the most common type of TV service. In this technology the TV stations broadcast TV content/signal to consumers/TV receivers having antenna, via radio waves from the terrestrial transmitter in digital format. This TV service allows the transmission of more number of standard TV channels, HD channels and even radio services to the consumers as compared to the old analogue TV service. Doordarshan, the national television channel of India has a network of fourteen hundred terrestrial transmitters which is one of the biggest terrestrial networks in the world. This explains the practical significance of Gray codes in digital communication networks In fact Gray codes facilitate error correction in some cable TV systems too.

2 Objectives of the paper

The problem of Gray codes construction using mathematical methods is discussed in this paper. The problem of generating Gray codes of any given length using a direct approach via Hamiltonian graphs and an iterative approach as in literature [4] has been analyzed and compared. The paper is divided into following sections. Section 3 discusses the preliminary knowledge required for Gray code construction. Section 4 discusses a direct method and an iterative the construction of Gray codes. Section 5 gives a conclusive analysis of the two methods discussed for generation of Gray codes followed by a list of references.

3 Preliminaries

3.1 Hamiltonian cycles and related results

In this subsection, we describe Hamiltonian cycles and some related results. Although strong results in regard to Hamiltonian cycles are not available generally, strong in the sense that those results which give if and only if conditions for a Hamiltonian cycle to be a part of a graph are not very well known. In fact testing a graph for the presence a Hamiltonian cycle is a computational problem for which there are no systematic algorithms or it is a NP complete problem. Still there are some known properties which come to rescue in finding out Hamiltonian cycles in a graph.

3.1.1 Some terminologies

1. A path in a graph G is a sequence of edges in G that begins at some vertex v of G and travels along the edges from one vertex of G to another. In figure 1, the sequence of edges e_1, e_2, e_4 is a path from v_1 to v_5.

2. A cycle in a graph G is a path in G which starts and ends at the same vertex and all the rest vertices and edges in such a path are distinct. In figure 1, the sequence of edges e_1, e_2, e_4, e_5, e_6 is a cycle from vertex given to vertex v_1 to v_1.

3. In a graph G, a Hamiltonian cycle is a cycle that contains each vertex of G. A graph G is a Hamiltonian graph if it has a Hamiltonian cycle. In figure 2 and figure 3, graph G_1 is a Hamiltonian graph as it has a Hamiltonian cycle, namely, a cycle with vertices $v_1, v_2, v_3, v_4, v_5, v_1$ while graph G_2 is not Hamiltonian as it does not have any Hamiltonian cycle.

4. In a graph G for a vertex ‘v’, the degree of ‘v’ is the count of edges incident on ’v’. In figure 2, the degree of vertex $v_1 = \deg(v_1) = 3$

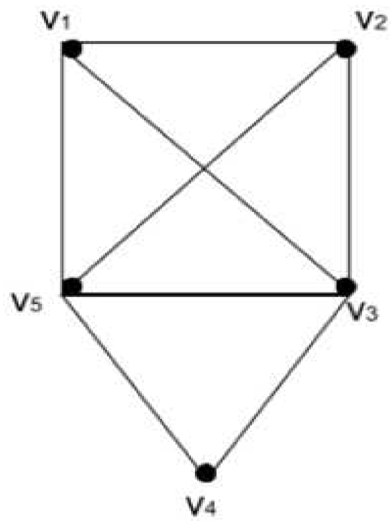

Fig. 2: **Graph** G_1

3.1.2 Some important results

1. (Dirac's theorem) A simple graph G (that is a graph without loops and multiple edges) with at least three vertices so that deg (v) is at least $\left(\frac{n}{2}\right)$, for every vertex 'v' in G, where 'n' being the count of vertices in G, contains a Hamiltonian cycle.

2. (Ore's theorem) A graph G (simple) having three vertices or more so that for every non-adjacent vertices 'u' and 'v' in G(i.e., vertices not joined by an edge) with deg(u) + deg(v) $\geq$ n , 'n' being the number of vertices in G, contains a Hamiltonian cycle.

3. The only cycle which is a part of a Hamiltonian cycle H of a graph G is H itself.

4. For any Hamiltonian cycle H in a graph G, for every vertex 'v' in G such that deg(v)=2; both the edges incident with 'v' are a part of H.

These are a few results related to the Hamiltonian graphs. More of such results are available in literature [5], [6].

3.2 Gray codes

A length 'n' Gray code is an ordered collection of 2^nsequences consisting of 'n' bits of '0' and '1' such that each sequence in the list, differs from the next one in exactly one bit (or one digit) and also the last sequence differs from the first one in one bit.
Example: - A length 2 Gray code consists of sequences:
00, 01, 11, 10
A Gray code of length 3 is the list of sequences:
000, 010, 011, 111, 110, 100, 101, 001

Fig. 3: **Graph** G_2

4 Constructing Gray codes

Constructing Gray codes is a problem of great concern while designing switching circuits with the objective of improving their dependability. Gray codes and their construction has been discussed with different approaches in literature [7], [8], [9].The approaches for construction of Gray codes used in this paper are as in literature [4]. A Gray code can be constructed by a direct method using Hamiltonian cycle in a Hamiltonian Graph. A Gray code can alternatively be constructed using an iterative method of generation. Both the approaches for construction of Gray codes as in [4] have been illustrated and compared for Gray codes of various lengths in the following subsections

4.1 Constructing Gray codes via Hamiltonian cycles

In this subsection using the approach as in [4] it has been shown that the problem of constructing a Gray code of a given length reduces down to the direct graphical problem of looking for a Hamiltonian cycle in the corresponding Hamiltonian graph.
A length 'n' Gray code can be visualized as a Hamiltonian cycle in a Hamiltonian graph G having 2^n vertices labeled with all possible length 'n' sequences of bits '0' and '1', where an edge in G is present between those vertices of G, whose corresponding labels differ in just one bit. We explicitly construct Gray code of lengths 2, 3 and 4

4.1.1 Length 2 Gray code

Here length 'n' of Gray code is 2.The following are the steps for construction:
Step 1: Construct a graph G with $2^n = 2^2 = 4$ vertices, with vertices labeled as the four possible length 2 sequences of bits '0' and '1'
Step 2: Join those vertices of G by an edge, for which the corresponding labels differ by

exactly one bit
Step 3: The graph G thus formed in figure 4 is a Hamiltonian graph. Find a Hamiltonian cycle in G by passing through the vertices in G, say, v_1, v_3, v_2, v_4, v_1
Step 4: Then writing the labels corresponding to the vertices in this Hamiltonian cycle gives the Gray code. Thus the length 2 Gray Code generated is: 00, 01, 11, 10

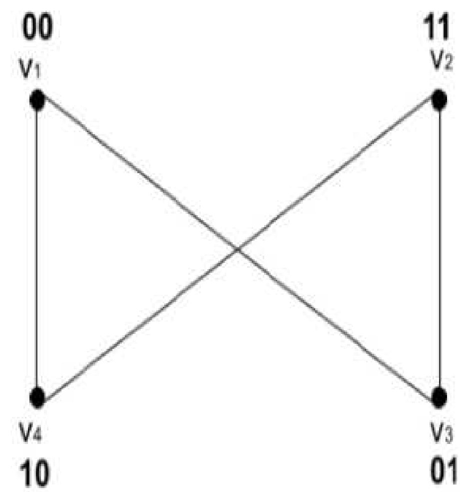

Fig. 4: **Graph for length 2 Gray code construction**

4.1.2 Length 3 Gray code

Here n= 3.Constructing first a graph G with $2^n = 2^3 = 8$ vertices, with vertices labeled as the eight possible length 3 sequences of bits '0' and '1'. Joining these vertices in G by an edge for which the corresponding labels differ by exactly one bit, we get a Hamiltonian graph G as shown in figure 5.

A Hamiltonian Cycle in this Hamiltonian graph formed gives the corresponding sequence of labels which forms a length 3 Gray Code given by: 000, 010, 011, 111, 110, 100, 101,001.

A better way of drawing the graph in figure 5 is shown in figure 6 where the graph is a cube. The darkened edges in the graph in figure 6 illustrate Hamiltonian cycle (Gray code)

4.1.3 Length 4 Gray code

A length 4 Gray code can be constructed in a similar manner using a Hamiltonian cycle in the corresponding graph of 16 vertices. A Gray code of length for 4 is: 0000, 0001, 0011, 0010, 0110, 0111, 0101, 0100, 1100, 1101, 1111, 1110, 1010,1011, 1001, 1000.The corresponding Hamiltonian graph G is shown in figure 7. Another length 4 Gray code

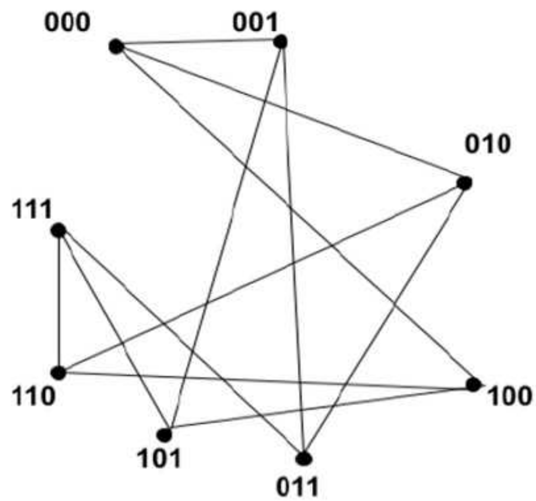

Fig. 5: **Graph for length 3 Gray code construction**

obtained using the labels of another Hamiltonian cycle in the figure 7 of Hamiltonian graph is: 0100, 0101, 0111, 0110, 0010, 0011, 0001, 0000, 1000, 1001, 1011, 1010, 1110, 1111, 1101, 1100.
Thus we get a method of construction of a length 'n' Gray code, for any 'n' using a Hamiltonian graph G having 2^n vertices.

(Source of figure 7: *https://en.wikipedia.org/wiki/File:Gray_code_tesseract.svg*)

Remarks:

1. The above illustration indicates that as in a Hamiltonian graph there may be more than one Hamiltonian cycle, so correspondingly Gray code of a given length generated via the Hamiltonian graph approach need not be unique.
2. The method discussed before also gives a method of constructing Hamiltonian graphs with 2^nvertices. For a given value of 'n' the Hamiltonian graph constructed in this manner is the unique Hamiltonian graph with 2^nvertices induced by any Gray code of length 'n'.

4.2 Construction of Gray codes via iterative method

The iterative algorithm for generating a length 'n+1' Gray code using a length 'n' Gray code as in [4] is as follows:
Steps of general iterative algorithm

1. Given a list with name, say, *list* of all 2^n sequences in a Gray code of length 'n; form two lists, namely, *list1* and *list2* using *list*, by following step 2 and step 3 respectively.

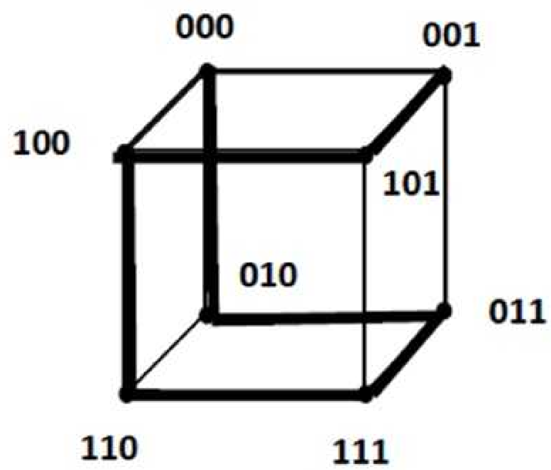

Fig. 6: **An alternate graph for length 3 Gray code construction**

2. *list1* is formed from *list* by appending a '0' in the beginning of each sequence of *list*.
3. *list2* is formed from *list* by appending a '1' in the beginning of each sequence of *list*.
4. Now a new list, namely, *list3* is formed from *list2* by writing the sequences in *list2* in a reverse order.
5. Concatenating or joining *list1* with *list3* gives a list of all $2^n + 2^n = 2^{n+1}$ sequences in a length 'n + 1' Gray code.

4.2.1 Generating length 2 Gray code from length 1 Gray code

An illustration of general iterative algorithm for generating a length '2' Gray code from a length '1' Gray code is given as follows:

1. Given *list* of sequences in a length 1 Gray Code i.e. 0, 1
2. Form *list1* by appending 0 before each sequence in *list*. So *list1* is: 00, 01
3. Form *list2* by appending 1 before each sequence in the *list*. So *list2* is: 10, 11
4. Form *list3* by reversing the order of elements in *list2*.So *list3* is: 11, 10
5. Concatenate *list1* and *list3* to get a list of 4 sequences representing a length 2 Gray code as follows: 00, 01, 11, 10

Remarks:

1. Alternatively, we can generate a length 2 Gray code from a length 1 Gray code by considering the Gray code 1, 0. This can be done as follows:

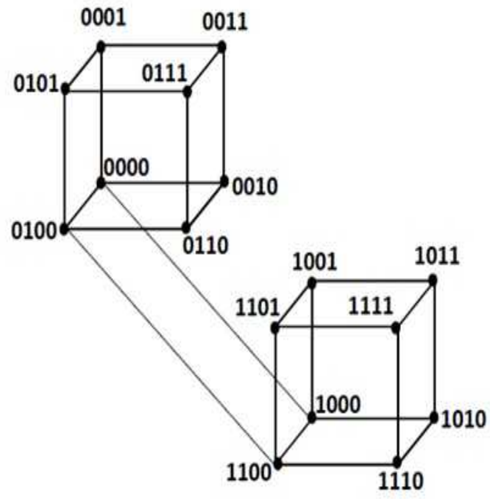

Fig. 7: **Graph for length 4 Gray code construction**

(a) Given *list* of sequences in a length 1 Gray Code i.e. 1, 0
(b) Form *list1* by appending 0 before each sequence in *list*. So *list1* is: 01, 00
(c) Form *list2* by appending 1 before each sequence in the *list*. So *list2* is: 11, 10
(d) Form *list3* by reversing the order of elements in *list2*.So *list3* is: 10, 11
(e) Concatenate *list1* and *list3* to get a list of 4 sequences representing length 2 Gray code as follows: 01, 00, 10, 11

2. It can therefore be seen that a length 2 Gray code is not unique. Also the length 2 Gray code generated by the iterative procedure depends upon the length 1 Gray code used to generate it.

4.2.2 Generating length 3 Gray code from length 2 Gray code

An illustration of the general iterative algorithm for generating a length '3' Gray code from a length '2' Gray code is given as follows:

1. Given *list* of sequences in a length 2 Gray Code i.e. 00, 01, 11, 10
2. Form *list1* by appending 0 before each sequence in *list*. So *list1* is: 000, 001, 011, 010
3. Form *list2* by appending 1 before each sequence in the *list*. So *list2* is: 100, 101, 111, 110
4. Form *list3* by reversing the order of elements in *list2*.So *list3* is: 110, 111, 101, 100
5. Concatenate *list1* and *list3* to get a list of 8 sequences representing a length 3 Gray code as follows: 000, 001, 011, 010, 110, 111, 101, 100

4.2.3 Generating length '4' Gray code from length '3' Gray code

For generating a length '4' Gray code from a length '3' Gray code iteratively, the method is as follows:

1. Given *list* of sequences in length 3 Gray Code i.e.000, 010, 011, 111, 110, 100, 101, 001
2. Form *list1* by appending 0 before each sequence in *list*. So *list1* is: 0000, 0010, 0011, 0111, 0110, 0100, 0101, 0001
3. Form *list2* by appending 1 before each sequence in the *list*. So *list2* is: 1000, 1010, 1011, 1111, 1110, 1100, 1101, 1001
4. Form *list3* by reversing the order of elements in *list2*.So *list3* is: 1001, 1101, 1100, 1110, 1111, 1011, 1010, 1000
5. Concatenate *list1* and *list3* to get a list of 16 sequences representing a length 4 Gray code as follows: 0000, 0010, 0011, 0111, 0110, 0100, 0101, 0001, 1001, 1101, 1100, 1110,1111,1011,1010, 1000

5 Conclusion

The two methods for Gray code construction illustrated in section 4 for generating Gray codes of various lengths are as discussed in literature [4].These two methods, namely the direct method of construction using the Hamiltonian graph approach and the iterative method have been analyzed and compared with each other to reach at the following conclusions.

1. It can be seen that as Hamiltonian cycles in a Hamiltonian graph may not be unique, so correspondingly the Gray code of a given length generated via the Hamiltonian graph approach need not be unique. Although the Hamiltonian Graph for a Gray code of any given fixed length 'n' is unique but the Gray code of length 'n' generated from the constructed Hamiltonian graph of 2^n vertices would depend upon the Hamiltonian cycle chosen in this constructed Hamiltonian graph. Also for the iterative method for Gray code generation, the Gray code of length 'n' that is generated is not unique. As remarked earlier in subsection 4.2, it can be seen that the length 'n' Gray code generated by the iterative procedure depends upon the length 'n-1' Gray code chosen to generate it.
2. The method of construction of a length 'n' Gray code using a Hamiltonian cycle in a Hamiltonian graph G having 2^n vertices illustrated in subsection 4.1,also gives a method for constructing Hamiltonian graphs with 2^n vertices. For a given value of 'n' the Hamiltonian graph constructed in this manner is unique and therefore the graph generated via this approach is the unique Hamiltonian graph with 2^n vertices induced by any Gray code of length 'n'.
3. The iterative method suggested for generating a length 'n+1' Gray code requires a length 'n' Gray code. However, the Hamiltonian graph approach for construction of a length 'n' Gray code is direct in the sense that it does not depend on the any Gray code of smaller length.

4. The iterative method generates one length ‘n+1’ Gray code at a time using a given length ‘n’ Gray code , while the Hamiltonian graph approach generates as many length ‘n+1’ Gray codes as the number of Hamiltonian cycles in the given constructed Hamiltonian Graph with 2^{n+1} vertices.

5. The discussion in the paper indicates that the problem of Gray code construction finally narrows down to the task of searching for a Hamiltonian cycle in the corresponding Hamiltonian graph. The Hamiltonian graph approach for constructing of Gray codes is apparently better than the iterative approach for construction of Gray Codes as in the Hamiltonian approach no initial Gray code is required to start the construction of a Gray code of a given length. However, this point of comparison of the two methods needs further assessment as to the number of computational steps required in each method to generate a Gray code of given length. Computational implementation of the iterative procedure has been discussed in literature [10]

The Gray codes have immense applications such as in rotary encoder sensors, cryptanalysis, chaotic cryptosystem, electro optical switches, electro chemical signals [11]. The fact that changing from one decimal digit to the next decimal digit requires a change of just one bit when Gray codes are used as shown in the table in figure 8 ,makes Gray codes a popular choice for reducing the amount of switching.

Decimal Number	Binary Representation	Gray Code of length 3
0	000	000
1	001	001
2	010	011
3	011	010
4	100	110
5	101	111
6	110	101
7	111	100

Fig. 8: **Gray code representation of decimals**

References

[1] https://www.statisticshowto.com/hamiltonian-cycle/

[2] Alhalabi, W., Kitanneh, O., Alharbi, A. et al.(2016) : Efficient solution for finding Hamilton cycles in undirected graphs. Springer Plus 5, Vol.1192 .https://doi.org/10.1186/s40064-016-2746-8

[3] Lucal, Harold M. (1959) : Arithmetic operations for digital computers using a modified reflected binary, IRE transactions on electronic computers. Vol. EC-8 , No. 4, pp. 449-458.

[4] Goodaire, Edgar G. and Parmenter , Michael M.(2013) : Discrete mathematics with graph theory(3rd ed.). pp. 315-317 , PHI learning private limited.

[5] Goedgebeur, Jan., Meersman, B., and Zamfirescu,Carol T. (2020) : Graphs with few Hamiltonian Cycles. Mathematics of Computation, Vol. 89, pp. 965-991.

[6] Brinkmann, G., Coolsaet, K., Gedgebeur, J. and Melot, H. (2013) : House of Graphs: a database of interesting graphs, Discrete Appl. Math. Vol.161, No.1-2, pp. 311-314.

[7] Gilbert, E .N. (1958) : Gray Codes and path on the n- cube, Bell System Tech.J. Vol. 37, pp. 815-826.

[8] Vickers, V.E. and Silverman, J. (1980) : A technique for generating specialized Gray Codes, IEEE Trans. Comput . Vol. 29, pp. 329-331.

[9] Ali, Mohsin , Islam, Muhammad Nazrul and Foysal, A. (2009) : Algorithms for Generating Binary Reflected Gray Code Sequence: Time Efficient Approaches. Proceedings - 2009 International Conference on Future Computer and Communication, ICFCC 2009. 10.1109/ICFCC.2009.41.

[10] Ahmad, Afaq and Bait-Suwailam Mohammed (2009) : A less Complex Algorithmic Procedure for Computing Gray Codes. Journal of Engineering Research. Vol. 6, No.2 , pp. 12-19.

[11] https://www.scribd.com/document/271037388/Application-of-Gray-Code