

# A Complete Characterization of Magic Constants Arising from Distance Magic Graphs

Ravindra Pawar<sup>1,✉, </sup>, Tarkeshwar Singh<sup>1</sup>, Himadri Mukherjee<sup>1</sup>, Jay Bagga<sup>2</sup>

<sup>1</sup> Department of Mathematics, BITS Pilani K K Birla Goa Campus, Goa, India

<sup>2</sup> Department of Computer Science, Ball State University, Indiana, USA

## ABSTRACT

A positive integer  $k$  is called a magic constant if there is a graph  $G$  along with a bijective function  $f$  from  $V(G)$  to first  $|V(G)|$  natural numbers such that the weight of the vertex  $w(v) = \sum_{uv \in E} f(u) = k$  for all  $v \in V$ . It is known that all odd positive integers greater than or equal to 3 and the integer powers of 2,  $2^t$ ,  $t \geq 6$  are magic constants. In this paper, we characterize all positive integers that are magic constants and generate all distance magic graphs, up to isomorphism, of order up to 10.

*Keywords:* Magic constant, Distance magic graph, Backtracking Algorithm

## 1. Introduction

Throughout this article, we assume that graph  $G = (V, E)$  is a finite simple graph with  $V$  denoting the set of vertices and  $E$  the set of edges. For a vertex  $u$ , its neighborhood is given by  $N(u) = \{v \in V : uv \in E\}$ . A positive integer  $k$  is called a *magic constant* if there is a graph  $G$  along with a bijective function  $f : V(G) \rightarrow \{1, 2, \dots, |V(G)|\}$  such that the weight of the vertex  $w(v) = \sum_{uv \in E} f(u)$ , remains constant and equal to  $k$  for all  $v \in V$ . Such labeling  $f$  is called *distance magic labeling*. A graph equipped with distance magic labeling is known as a *distance magic graph* (for more details, see [1], [4]). We refer to West[15] for graph-theoretic terminology and notation not covered here.

Regardless of how the vertices are labeled, the magic constant, if it exists, remains invariant (see [2, 11]); that is, it is independent of the distance magic labeling. At the International Workshop on Graph Labeling (IWOGL-2010), Arumugam [1] raised the question to *characterize the set of positive integers, that are magic constants*. Since then, this problem has captured the attention of

✉ Corresponding author.

*E-mail addresses:* [p20200020@goa.bits-pilani.ac.in](mailto:p20200020@goa.bits-pilani.ac.in) (R. Pawar), [tk Singh@goa.bits-pilani.ac.in](mailto:tk Singh@goa.bits-pilani.ac.in) (T. Singh), [himadrim@goa.bits-pilani.ac.in](mailto:himadrim@goa.bits-pilani.ac.in) (H. Mukherjee), [jbagga@bsu.edu](mailto:jbagga@bsu.edu) (J. Bagga).

Received 03 August 2024; accepted 05 December 2024; published 31 December 2024.

DOI: [10.61091/jcmcc123-20](https://doi.org/10.61091/jcmcc123-20)

© 2024 The Author(s). Published by Combinatorial Press. This is an open access article under the CC BY license (<https://creativecommons.org/licenses/by/4.0/>).

researchers. The integers 1, 2, 4, 6, 8, 12 do not belong to the set of magic constants. On the other hand, all odd integers  $\geq 3$  and all integers of the form  $2^t$  ( $t \geq 6$ ) are confirmed members of the set of magic constants (see, [7], [8]). A previous study in [16] identified a graph with 8 vertices that admits the magic constant 24, and it is straightforward to verify that 10 is the magic constant for the wheel graph  $W_5$  [7]. However, characterizing the set of integers, which are magic constants, is an ongoing problem in distance magic graphs. We have shown that positive integers of the form  $4t + 2$  for  $t \geq 3$  and of the form  $4t + 4$  for  $t \geq 8$  are magic constants. Hence, the remaining numbers are 16, 20, 28, and 32.

Bertault et al. [3] introduced a heuristic algorithm to identify various classes of labelings. Building on this, Fuad et al. [16] refined the algorithm, achieving the construction of all distance magic graphs up to isomorphism for orders up to 9. This is the well-known algorithm available for constructing distance magic graphs. However, it is heuristic, and from a computational point of view, this algorithm has limitations. It relies on a collection of all non-isomorphic graphs as input, a formidable task that poses challenges, particularly for higher-order graphs. Generating all non-isomorphic graphs of substantial sizes remains computationally demanding, thereby affecting the practicality of the algorithm.

In this paper, we solve the characterization problem of magic constants. We prove this result by providing constructions of distance magic graphs to obtain specific magic constants.

We have devised and implemented a backtracking algorithm to check if there is a distance magic graph for a given number of vertices  $n$  and a specific magic constant  $k$ . Our algorithm is deterministic, and it explores all the possible solutions in a robust manner. We have implemented this algorithm to construct magic graphs for  $k = 28$  and 32. Our search for  $k = 16$  shows that no graph admits the magic constant 16. Additionally, with this algorithm, we have generated all distance magic graphs (up to isomorphism) of order up to 12 (see Table 1). This enhances the existing collection of distance magic graphs given in [16]. In particular, we find a new sixth graph of order 8, adding to the list of 5 such graphs given in [16].

This algorithm serves as a practical tool for researchers and enthusiasts in the field of distance magic graphs. Addressing the general challenge of characterizing all graphs possessing distance magic labelings using algorithms presents a computational hurdle. This is mainly due to the impracticality of generating all non-isomorphic graphs since this becomes increasingly unfeasible as the graph order grows. Consequently, the algorithms devised for characterization problems often encounter limitations or become computationally intensive when dealing with graphs of larger vertices.

It is believed that for a given  $n$ , only a small subset of the set of all graphs of order  $n$  possesses the distance magic property. Consequently, a different approach is required rather than providing a collection of non-isomorphic graphs as input and subsequently characterizing them for the distance magic property. Instead, the focus shifts towards developing algorithms capable of dynamically constructing all distance magic graphs on  $n$  vertices. By adopting this strategy, we can bypass the challenge of generating all non-isomorphic graphs. We have successfully implemented this approach.

## 2. Main Results

Let  $G$  be a graph. A subset  $S$  of the vertex set of  $G$  is said to be a *dominating set* if there is a function  $f : V \rightarrow \{0, 1\}$ , called *dominating function*, such that  $f(S) = \{1\}$ ,  $f(V \setminus S) = \{0\}$  and for

each vertex  $v \in V \setminus S$ ,  $\sum_{u \in N(v)} f(u) \geq 1$ . The *domination number*  $\gamma(G)$  of a graph is given by

$$\gamma(G) = \min\{|f|: f \text{ is a dominating function}\},$$

where  $|f| = \sum_{v \in V} f(v)$ . There is another variation of domination called fractional domination.

**Definition 2.1.** A function  $g : V(G) \rightarrow [0, 1]$  is called a *total dominating function* of the graph  $G$  if  $\sum_{u \in N(v)} g(u) \geq 1$  for all  $v \in V$ . The *fractional total domination number* of a graph  $G$  is denoted by  $\gamma_{ft}$  and is given by

$$\gamma_{ft} = \min\{|g|: g \text{ is a total dominating function of } G\},$$

where  $|g| = \sum_{v \in V} g(v)$ .

This concept of a fractional domination number has a close relation with the magic constant of a graph as stated in the following theorem.

**Theorem 2.2.** [2, 10] If a graph  $G$  is distance magic, then its distance magic constant  $k = \frac{n(n+1)}{2\gamma_{ft}}$ .

**Theorem 2.3.** [6, 9, 13, 14] Let  $G$  be a distance magic graph on  $n$  vertices. If  $G$  is distance magic with distance magic labeling  $f$  and magic constant  $k$  then  $\sum_{v \in V} w(v) = nk$ .

**Corollary 2.4.** If  $G$  is an  $r$ -regular distance magic graph on  $n$  vertices with magic constant  $k$ , then  $k = \frac{r(n+1)}{2}$ .

**Theorem 2.5.** [8] There exists a 4-regular distance magic graph on an odd number of vertices  $n$  if and only if  $n \geq 17$ .

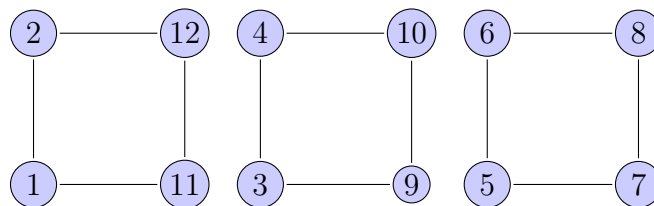


Fig. 1. A distance magic labeling of  $3C_4$

It is known that a 2-regular graph  $G$  is distance magic if and only if it is a disjoint union of copies of cycle  $C_4$  [6]. Further, each positive integer of the form  $4t + 1$  is a magic constant of a graph union of  $t$  copies of a 4-cycle  $tC_4$  as stated in the following theorem.

**Theorem 2.6.** [6] A graph  $G$  of order  $n$  is a distance magic graph with magic constant  $k = n + 1$  if and only if  $G = tC_4$ ,  $t \geq 1$ .

The following result shows that each positive integer of the form  $4t + 3$  is a magic constant of a graph  $P_3 \cup tC_4$ .

**Theorem 2.7.** [6] Let  $G$  be a magic distance graph with magic constant  $k$ . Then, the following are equivalent.

- (a)  $k = n$ .
- (b)  $\delta(G) = 1$ .
- (c) Either  $G$  is isomorphic to  $P_3$  or  $G$  contains exactly one copy of  $P_3$  and all other components are isomorphic to  $C_4$ .

Figure 2 shows the distance magic labeling of  $P_3 \cup 2C_4$  with magic constant 11.

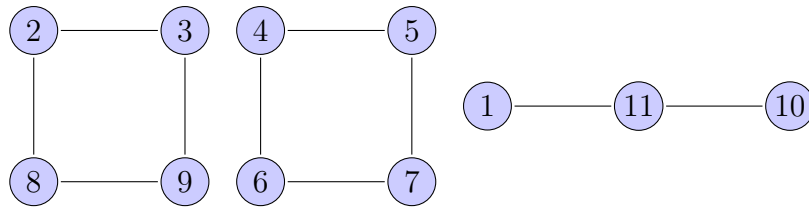


Fig. 2. A distance magic labeling of  $P_3 \cup 2C_4$

From Theorems 2.6 and 2.7, we conclude that all odd positive integers  $\geq 3$  are magic constants. This also answers the question: *does  $p^t, t \geq 1$  belong to the set of magic constants, where  $p$  is odd prime?* (see [7]).

Now, we explore the case of the even positive integers, which are magic constants.

When  $H$  is an arbitrary graph with vertices  $x_1, x_2, \dots, x_n$ , and  $G$  is any graph with  $t$  vertices, then by  $H[G]$  we denote the graph, which arises from  $H$  by replacing each vertex  $x_i$  by a copy of the graph  $G$  with vertex set  $X_i = \{x_{i1}, \dots, x_{it}\}$ , and each edge  $x_i x_j$  by the edges of the complete bipartite graph  $K_{t,t}$  with bipartition  $X_i, X_j$ . The graph  $H[G]$  is then called *lexicographic product* or the *composition* of  $H$  and  $G$ .

**Theorem 2.8.** [5, 9] If  $G$  is an  $r$ -regular graph, then  $G[\overline{K}_t]$  is distance magic for any even  $t$ .

In [9], the authors proved that for an  $r$ -regular graph  $G$  on  $n$  vertices, the magic constant of  $G[\overline{K}_{2t}]$  is  $k = rt(2nt + 1)$ . With  $r = 2$  and  $t = 1$ , we obtain  $k = 4n + 2$ . This gives the proof of the following theorem.

**Theorem 2.9.** For all  $t \geq 3$ ,  $4t + 2$  is a magic constant.

In [8], the authors proved that there exists a 4-regular distance magic graph  $G = (V, E)$  of odd order if and only if  $|V| \geq 17$ . Now, we use this result to prove the existence of even magic constants of the form  $4t + 4, t \geq 8$ .

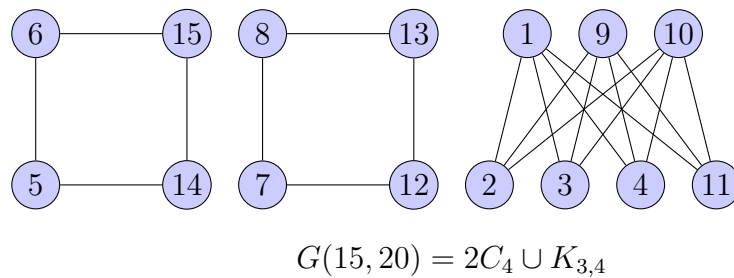
**Theorem 2.10.** For all  $t \geq 8$ ,  $4t + 4$  is a magic constant.

**Proof.** Let  $G$  be a 4-regular distance magic graph of odd order. Then,  $|V(G)| \geq 17$ . By Corollary 2.4, the magic constant  $k$  of this graph  $G$  is given by  $k = \frac{4(|V|+1)}{2} = 2|V|+2$ . If we write  $|V| = 2t + 1$ , where  $t \geq 8$ , then  $k = 4t + 4$ . This proves the theorem.  $\square$

**Theorem 2.11.** No regular graph admits magic constant 16.

**Proof.** We know that, if 16 is a magic constant of an  $r$ -regular graph  $G$  on  $n$  vertices then by Corollary 2.4,  $k = \frac{r(n+1)}{2}$  and  $n \leq k$ . This implies  $n = \frac{32}{r} - 1$ . Since  $n$  is an integer, we have  $r = 1, 2, 4, 8, 16$ , or  $32$ . Since  $G$  is  $r$ -regular graph on  $n$  vertices, we must have  $n > r$ . Therefore, we discard the possibilities  $r = 8, 16$  and  $32$ . With  $r = 4$  we get  $n = 7 < 17$ . Therefore, by Theorem 2.5,  $r = 4$  is not possible, and hence we discard this possibility. There cannot be 1-regular distance magic graph on 32 vertices. Therefore, we discard the case  $r = 1$ . Recall that, if  $r = 2$  then  $G$  is union of disjoint copies of  $C_4$ . Hence, by Theorem 2.6, the magic constant  $k \equiv 1 \pmod{4}$ . But  $k = 16$  and  $16 \not\equiv 1 \pmod{4}$ . Therefore, we discard the case  $r = 2$ . This completes the proof.  $\square$

We already know that the integers 1, 2, 4, 6, 8 and 12 are not magic constants [7], while 10, 20 [7], and 24 [16] are magic constants, as shown in Figures 3, 6, 8, respectively. Also, we will see in Section 3 that these distance magic graphs can also be generated using our algorithm with the given magic constants. Theorems 2.9 and 2.10 provide further insights into the nature of the set of magic constants. Theorem 2.9 reveals that all positive integers congruent to 2 (mod 4), with the exceptions of 2 and 6, are the members of the set of magic constants. Meanwhile, Theorem 2.10 establishes that all positive integers greater than or equal to 36 and congruent to 0 (mod 4) are also in the set of magic constants.



**Fig. 3.** Distance magic graph with magic constant 20

As a result, the only remaining even positive integers requiring examination as magic constants are 16, 28, and 32. We devise an algorithmic approach to determine whether these integers qualify as magic constants of some graphs. Thus, we obtain a complete characterization of all magic constants (see Theorem 5.1).

Consider the following example given in [12]: If the generalized Mycielskian of the complete bipartite graph  $\mu_m(K_{a,b})$  is distance magic, then by Theorem 2.2, its magic constant is given by  $k = \frac{(m(a+b)+a+b+1)(m(a+b)+a+b+2)}{2m+3}$ . When  $a = b = 1$ , then  $k = 2m + 4$ , which is an integer, but  $K_{1,1} \cong K_2$  and  $\mu_m(K_2)$  is not distance magic. On the other hand, for the case where  $a = b = 2$  and  $m \geq 1$ ,  $k$  is an integer  $8m + 10$  and  $\mu_m(K_{2,2})$  is distance magic. When  $k$  is not an integer, Theorem 2.2 implies that  $G$  is not distance magic. However, when  $k$  is an integer  $G$  may or may not be distance magic.

Therefore, it is noteworthy that given a graph  $G$  and the exact value of  $\gamma_{ft}(G)$ , one can compute the value of (the magic constant)  $k$ , but it is not possible to decide whether the underlying graph  $G$  is distance magic or not using Theorem 2.2. Even if the fraction  $\frac{n(n+1)}{2\gamma_{ft}(G)}$  is an integer, the graph  $G$  may or may not be distance magic.

The algorithm that we describe below is a natural way to construct distance magic graphs, and it has been adopted previously for smaller order graphs (see Chapter 2, [7]). Here, we give a general and robust version of the same.

### 3. A Backtracking Algorithm for Generating Distance Magic Graphs and Magic Constants

Given positive integers  $n$  and  $k$ , our algorithm checks if there exists a distance magic graph with vertex set  $\{1, 2, \dots, n\}$  and a magic constant  $k$ . We denote such graphs as  $G(n, k)$ . Without loss of generality, we may assume that each vertex  $i$  is labeled as  $i$  for  $1 \leq i \leq n$ . The algorithm returns the first successful search if such a distance magic graph exists.

Let  $S$  be the set of all subsets of the set  $\{1, 2, \dots, n\}$  with the sum of its elements equal to  $k$ . For each  $1 \leq i \leq n$ , let  $NS(i) = \{T \in S : i \notin T\}$ . Note that if there exists a distance magic graph  $G(n, k)$ , then the neighborhood of a vertex  $i$ ,  $N(i)$ , is a member of  $NS(i)$ . Next, fix a neighborhood of a vertex  $n$ , that is, an element of  $NS(n)$ , and construct a tree rooted at  $N(n)$  in the following manner. At level 0, we have only the root node. Add all elements in  $NS(n - 1)$  as children to  $N(n)$ . This is level 1. Add elements of  $NS(n - 2)$  as children to each node in level 1. This is level 2. Continue until we add all elements in  $NS(1)$  to each node in level  $n - 1$ . Note that at level  $i \geq 1$ , each node of this tree  $T(n, k)$  is a possible candidate for a  $N(n - i)$ . Each branch of this tree  $T(n, k)$  gives an adjacency list for the vertices  $1, 2, \dots, n$ . These lists may or may not correspond to a graph. Whenever adjacency lists corresponding to a branch give a graph, we call such a branch a *successful* branch.

#### 3.1. Algorithm

The algorithm consists of the following steps:

- Step 1. Generate all subsets of the set  $\{1, 2, \dots, n\}$  that have the sum of its elements equal to  $k$ .
- Step 2. For each vertex  $1 \leq i \leq n$ , construct  $NS(i)$ .
- Step 3. Construct a rooted tree  $T(n, k)$  with a root as a fixed element of  $NS(n)$ .
- Step 4. For a tree  $T(n, k)$  obtained in Step (3), find a successful branch. Repeat Steps (3) and (4) for each element of  $NS(n)$  until we find a successful branch or run out of elements in  $NS(n)$ .

To optimize the search space, we merge Step (3) and Step (4) into a new streamlined Step (3'), as below.

- Step 3'. We explore the branches of  $T(n, k)$  as discussed in Step (3) and Step (4) with the following condition. Let the tree be constructed to the level  $i(0 \leq i \leq n - 2)$  and  $NS(n - i) = \{s_1, s_2, \dots, s_{t_i}\}$ ,  $NS(n - (i + 1)) = \{r_1, r_2, \dots, r_{t_h}\}$ . Next, while adding children  $r_b$  at level  $i + 1$  to any of the nodes  $s_a$  we will check the following two conditions.

$$n - (i + 1) \text{ is in any of its ancestors at level } j \leq i \text{ if and only if } n - j \in r_b. \tag{1}$$

and

$$\text{neighborhood sum of } s_a \text{ does not exceed } k. \tag{2}$$

The part (1) of the Step (3') ensures symmetry, which means that  $u$  is adjacent to  $v$  if and only if  $v$  is adjacent to  $u$ . The part (2) of Step (3') ensures that the neighborhood sum of any vertex at any stage does not exceed the magic constant  $k$ . Although this condition may not seem critical at this point, it plays a crucial role in controlling the growth of the tree (see Section 3.3). If at least one of the conditions (1) or (2) stated in Step 3' are not met at any step, we avoid adding the corresponding node, preventing further growth of the tree below that node. This reduction strategy significantly

reduces the order and therefore the size of the tree  $T(n, k)$ , leading to an exponential reduction in the search space.

Step (1) uses backtracking to generate subsets of  $\{1, 2, \dots, n\}$  of the given sum  $k$ . The Step (3') of the algorithm involves a backtracking strategy in the process of finding a successful branch in the constructed tree  $T(n, k)$ . The algorithm systematically explores the branches of each rooted tree, corresponding to a different choice of elements from the set  $NS(n)$ . If, during this exploration, it is determined that a particular branch cannot lead to success, the algorithm backtracks to the previous decision point and explores alternative branches. This backtracking mechanism allows for an exhaustive search of the solution space, ensuring that all possible combinations of elements are considered. The backtracking in Step (3') is crucial to efficiently navigate the complex solution space represented by the tree, avoiding unnecessary computations and ultimately identifying successful branches of  $T(n, k)$ .

We provide the pseudocode for the algorithm in the Appendix.

### 3.2. Proof of correctness

It is sufficient to prove that every  $G(n, k)$  graph corresponds to a (successful) branch of some  $T(n, k)$  and each successful branch of each  $T(n, k)$  corresponds to some graph  $G(n, k)$ . This follows by observing that if  $T(n, k)$  has a successful branch, then the nodes on that branch constitute a graph  $G(n, k)$ .

Conversely, suppose that there is a graph  $G(n, k)$ . Consider a tree  $T(n, k)$  rooted at some element  $r$  of  $NS(n)$ . There is a one-to-one correspondence between branches of  $T(n, k)$  and the elements of the cartesian product  $NS(n - 1) \times NS(n - 2) \times \dots \times NS(1)$ . The neighborhood  $N(i)$  of each vertex  $i$  is a member of  $NS(i)$ . Hence,  $(N(n - 1), N(n - 2), \dots, N(1))$  is an element of the cartesian product  $NS(n - 1) \times NS(n - 2) \times \dots \times NS(1)$ . Thus, it corresponds to some branch of  $T(n, k)$ . This establishes the correctness of the algorithm.

### 3.3. Illustration for $G(7, 7)$

Let us illustrate the steps of our algorithm with  $G(7, 7)$ . By Theorem 2.7, we know that there is only one graph  $P_3 \cup C_4$  up to isomorphism on 7 vertices admitting the magic constant 7. All subsets of the set  $\{1, 2, \dots, 7\}$  having sum 7 are:

$$[[7], [6, 1], [5, 2], [4, 3], [4, 2, 1]].$$

We have listed all the subsets as a list of lists for the proper indexing purpose in the pseudocode. This completes Step 1. Next, we generate all possible neighborhood sets for each vertex as suggested in Step 2.

$$\begin{aligned} NS(1) &= [[7], [5, 2], [4, 3]] \\ NS(2) &= [[7], [6, 1], [4, 3]] \\ NS(3) &= [[7], [6, 1], [5, 2], [4, 2, 1]] \\ NS(4) &= [[7], [6, 1], [5, 2]] \\ NS(5) &= [[7], [6, 1], [4, 3], [4, 2, 1]] \\ NS(6) &= [[7], [5, 2], [4, 3], [4, 2, 1]] \\ NS(7) &= [[6, 1], [5, 2], [4, 3], [4, 2, 1]] \end{aligned}$$

Now we construct a tree  $T(7, 7)$  rooted at the node  $[6, 1]$  from  $NS(7)$ . This makes  $N(7) = [6, 1]$ . Add all elements of the set  $NS(6)$  as children to the root node as shown in Figure 4. The set of possible candidates for  $N(6)$  is

$$NS(6) = [[7], [5, 2], [4, 3], [4, 2, 1]].$$

We have chosen  $NS(7) = [6, 1]$ , to maintain symmetry as stated in (1) of Step 3', we must have  $7 \in N(6)$ . The only such element in  $NS(6)$  is  $[7]$ . Hence, we discard all other nodes in the level 1, and the tree won't grow further below those nodes. Now we go to level 2. The set of possible candidates for  $N(5)$  is

$$NS(5) = [[7], [6, 1], [4, 3], [4, 2, 1]].$$

So far we have  $N(7) = [6, 1]$  and  $N(6) = [7]$ . Since  $5 \notin N(6) \cup N(7)$ ,  $6, 7 \notin N(5)$  maintaining the condition given in (1) of Step 3'. Then the only choices for  $N(5)$  are  $[4, 3]$  and  $[4, 2, 1]$ . Without loss of generality, consider  $N(5) = [4, 3]$ . Now we go to the level 3. The set of possible candidates for  $N(4)$  is

$$NS(4) = [[7], [6, 1], [5, 2]].$$

So far we have  $N(7) = [6, 1]$ ,  $N(6) = [7]$  and  $N(5) = [4, 3]$ . Here  $4 \notin N(6) \cup N(7)$  hence  $6, 7 \notin N(4)$  but  $4 \in N(5)$  hence  $N(4)$  must contain 5. The only choice for  $N(4)$  is  $[5, 2]$ . We continue in a similar way, and lastly, we add pendants  $[7]$ ,  $[5, 2]$ , and  $[4, 3]$  from the set  $NS(1)$ . For  $N(1)$  we discard the nodes  $[5, 2]$  and  $[4, 3]$  because  $1 \notin N(j)$  for any  $j = 2, 3, 4, 5$  and  $N(1)$  must contain 7 since  $1 \in N(7)$ , hence maintaining symmetry condition as given in (1) of Step 3'.

In Figure 4, we have illustrated a case of a successful branch (colored blue).

Now let us consider a case where we select the node  $[4, 2, 1]$  instead of  $[4, 3]$  at the level 2 (see the red colored branch in Figure 4). In this case, so far we have  $N(7) = [6, 1]$ ,  $N(6) = [7]$  and  $N(5) = [4, 2, 1]$ . Then we go to level 3. The set of possible candidates for  $N(4)$  is:

$$NS(4) = [[7], [6, 1], [5, 2]].$$

Since  $4 \notin N(6) \cup N(7)$  but  $4 \in N(5)$ , we must have  $6, 7 \notin N(4)$  but  $5 \in N(4)$ . The only such possibility is  $[5, 2]$ . But then  $4, 5 \in N(2)$  and the neighborhood sum of the vertex 2 is more than  $k = 7$ . This violates the condition (2) of Step 3'. Therefore,  $N(4)$  has no choices and we discard this unsuccessful branch. This discarded branch is shown in red color in Figure 4.

### 3.4. Graphs output with our algorithm

For a distance magic graph  $G$  on  $n$  vertices with distance magic labeling  $f$  and magic constant  $k$ , it is easily seen that  $\sum_{v \in V} deg(v)f(v) = nk$  [14]. For any vertex  $v$ ,  $1 \leq \delta(G) \leq deg(v) \leq \Delta(G) \leq n - 1$ . Hence, we have an upper bound  $k \leq \frac{n^2-1}{2}$ . By the definition of the distance magic graph,  $k$  cannot be less than  $n$ . Hence, we have  $n \leq k \leq \frac{n^2-1}{2}$ , and the lower bound is sharp. Therefore, to check whether a given positive integer  $k$  is a magic constant, we need to run the algorithm for each pair  $(n, k)$ , where  $1 \leq n \leq k$ . Let  $\alpha$  be the largest positive integer such that  $\frac{\alpha(\alpha+1)}{2} \leq k$ . Hence, running the algorithm for the pairs  $(n, k)$ , where  $\alpha \leq n \leq k$  is sufficient. For example, for  $k = 16$ , the possibilities for  $n$  are  $6, 7, \dots, 16$ . Further, we can omit a few values in some cases, as mentioned in (Chapter 2, [7]).

Our algorithm explores the possibility of  $k = 16$  being the magic constant for the graphs on a possible number of vertices  $n$ . However, findings show that no graph admits the magic constant 16.



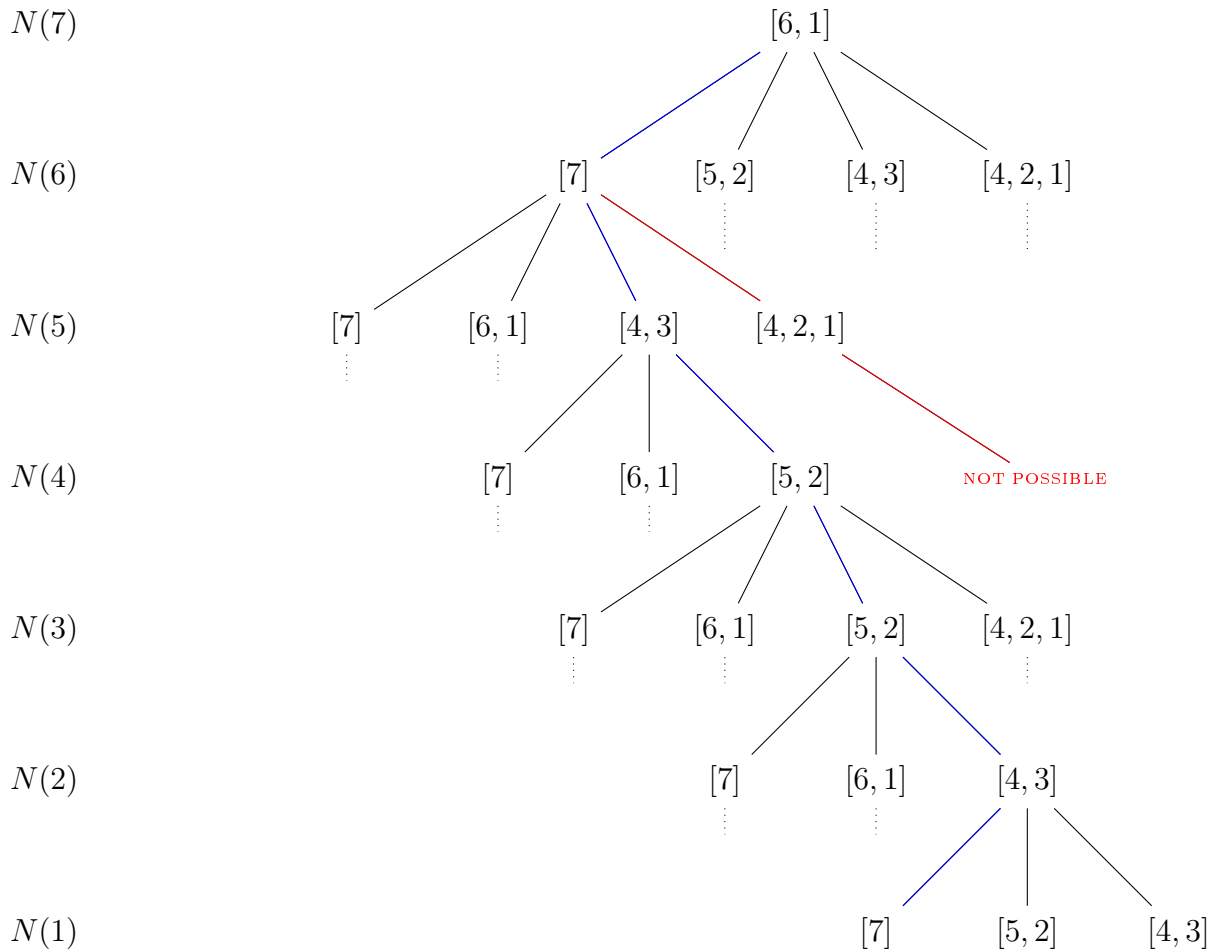


Fig. 4. A tree  $T(n, k)$  rooted at  $[6, 1]$  with a successful branch

The algorithm generates multiple graphs that admit 28 and 32 as magic constants. As a representative example, we list one of the such graphs in Figure 5. Due to the large number of non-isomorphic distance magic graphs of order 11 and 12 (see Table 1), we will list all non-isomorphic distance magic graphs of order up to 10 (see Section 4).

number of vertices: $n$	3	4	5	6	7	8	9	10	11	12
number of non-isomorphic distance magic graphs	1	1	1	1	3	6	5	5	74	1160

Table 1. Number of non-isomorphic distance magic graphs of order up to 12

### 4. Distance magic graphs of order up to 10

In this section, we list all non-isomorphic distance magic graphs of order up to 10 in Figures 6–10.

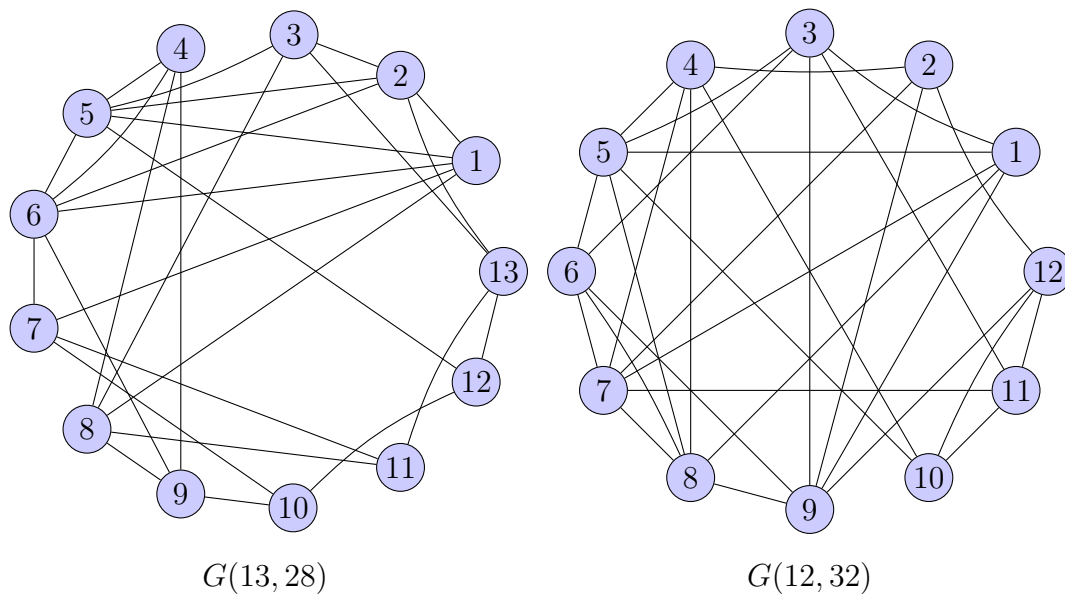


Fig. 5. Distance magic graphs with magic constants 28 and 32

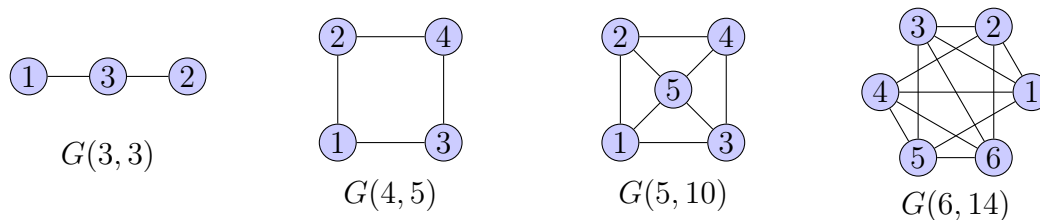


Fig. 6. Non-isomorphic distance magic graphs of order up to 6

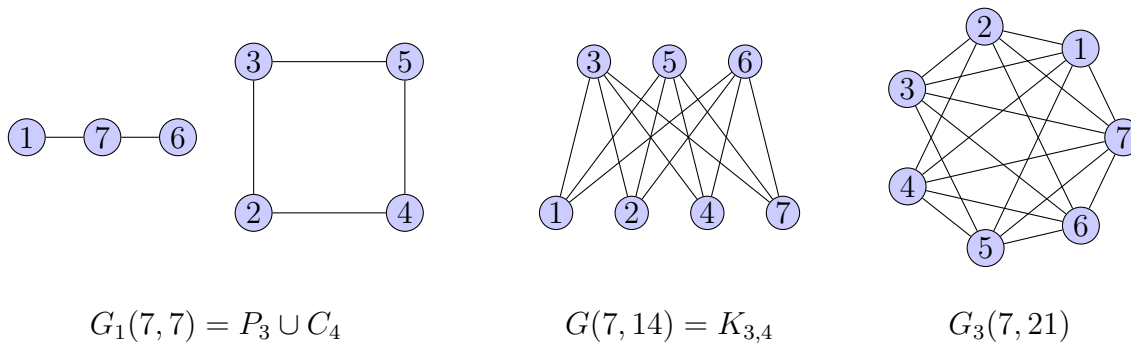


Fig. 7. Non-isomorphic distance magic graphs of order 7

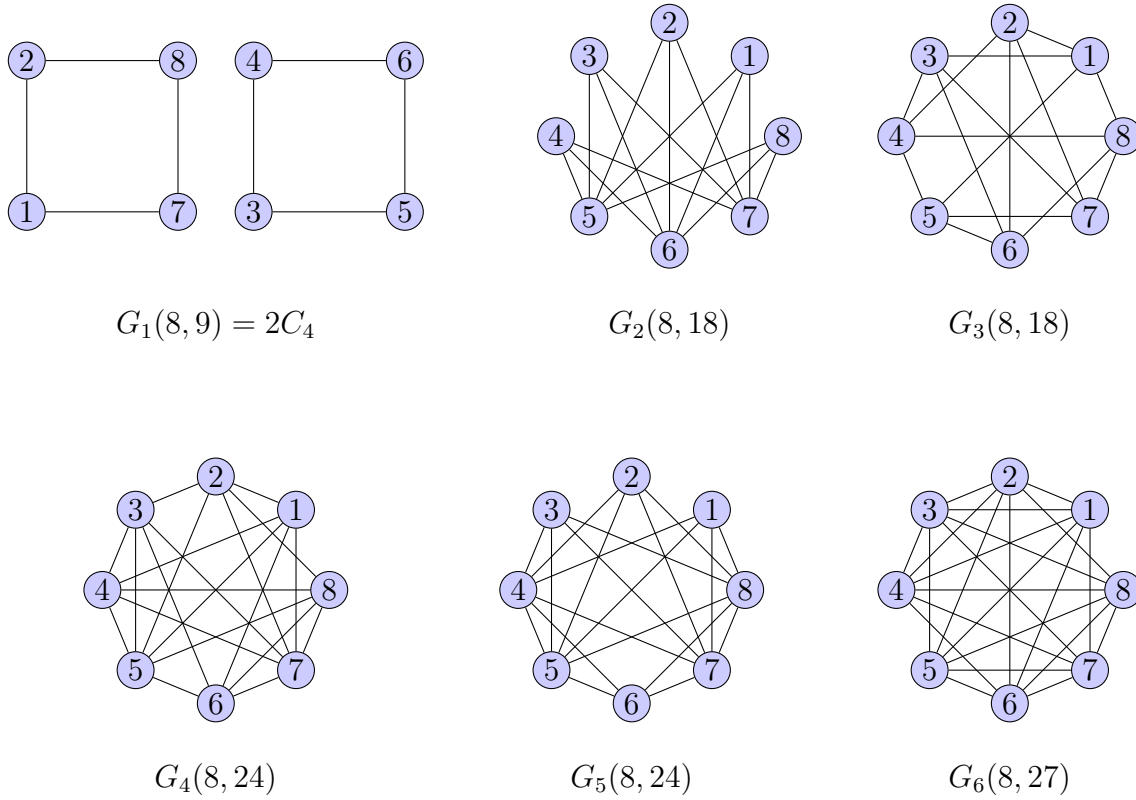


Fig. 8. Non-isomorphic distance magic graphs of order 8

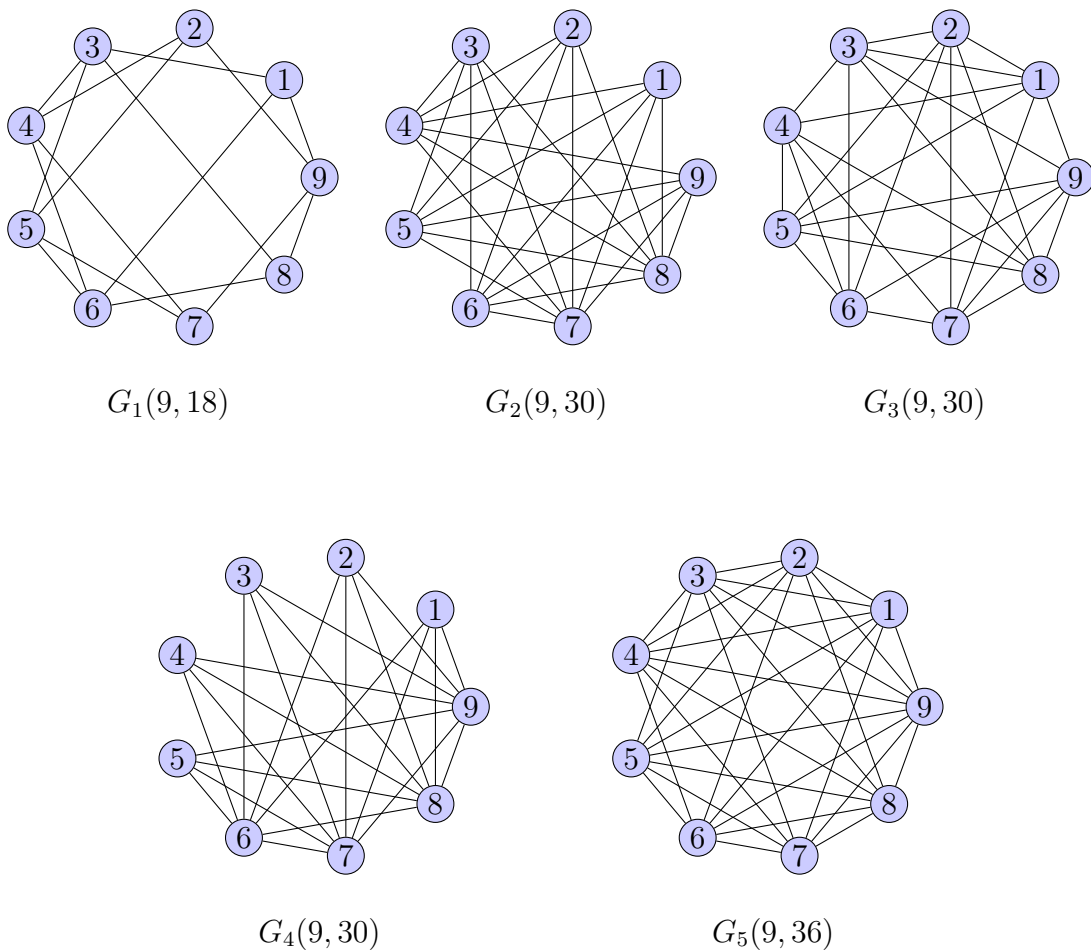
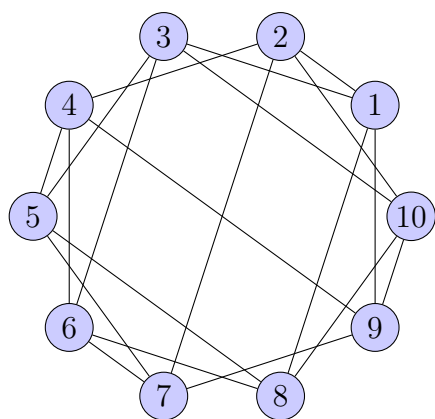
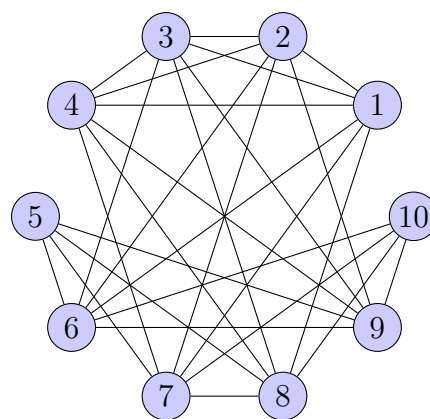
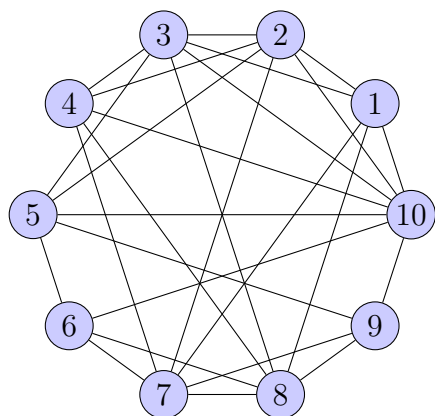
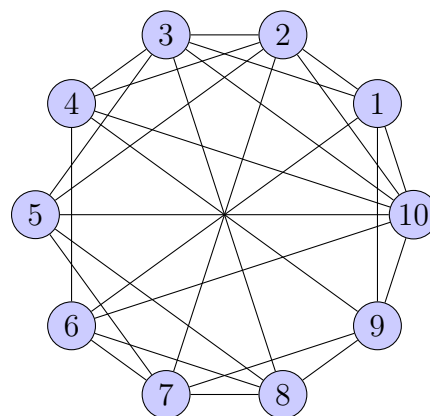
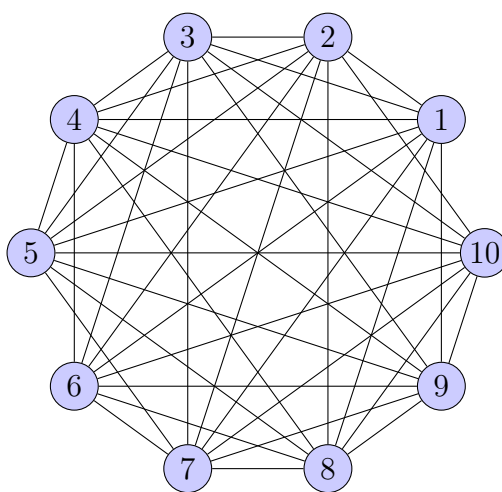


Fig. 9. Non-isomorphic distance magic graphs of order 9

 $G_1(10, 22)$  $G_2(10, 30)$  $G_3(10, 30)$  $G_4(10, 30)$  $G_5(10, 44)$ **Fig. 10.** Non-isomorphic distance magic graphs of order 10

## 5. Conclusions

There is no known polynomial-time algorithm to generate distance magic graphs on a given number of vertices and a given magic constant. Our algorithm presented in Section 3 is practically implementable for considerably large order distance magic graphs. This algorithm can be extended to generate  $D$ -distance magic graphs introduced by O'Neal et al. [10]. Let  $G$  be a graph and let  $\text{diam}(G)$  denote the diameter of  $G$ . Let  $D \subset \{1, 2, \dots, \text{diam}(G)\}$  and let  $W$  be a multiset of positive real numbers of size equal to the number of vertices in  $G$ . A graph  $G$  is said to be  $D$ -distance magic if there is a bijection  $f : V(G) \rightarrow W$  such that  $w(x) = k$  for all vertices  $x$ , where  $w(x) = \sum_{\{y:d(y,x) \in D\}} f(y)$  and  $d(x, y)$  denotes the distance between  $x$  and  $y$ .

With the results known earlier and those verified in this paper, the magic constants are now completely characterized, and we have the following theorem.

**Theorem 5.1.** *All positive integers except 1, 2, 4, 6, 8, 12, and 16 are magic constants.*

Not all magic constants arise from connected graphs. Also, for some integers, for example,  $n = k = 7$ , there is a unique distance magic graph  $G(n, k)$ . This naturally leads to the following questions.

- (a) Which magic constants are realized by connected distance magic graphs?
- (b) For which pairs  $(n, k)$ , there is unique distance magic graph on  $n$  vertices with magic constant  $k$ ?

We conclude by observing that it would be interesting to find a non-algorithmic proof for the case  $k = 16$ .

## Acknowledgements

The authors are grateful to Atharva Karandikar and Satyaprasad for their invaluable assistance in implementing the algorithm.

## References

- [1] S. Arumugam, D. Froncek, and N. Kamatchi. Distance magic graphs—A survey. *Journal of the Indonesian Mathematical Society*, (Special edition):11–26, 2011.
- [2] S. Arumugam, N. Kamatchi, and G. R. Vijayakumar. On the uniqueness of  $D$ -vertex magic constant. *Discussiones Mathematicae. Graph Theory*, 34(2):279–286, 2014. [10.7151/dmgt.1728](https://doi.org/10.7151/dmgt.1728).
- [3] B. François, F. Raquel, M. Miller, P. Helmut, and V. Ehsan. A heuristic for magic and antimagic graph labellings. In *Proc. VII Spanish Congress on Metaheuristics and Evolutionary and Bioinspired Algorithms 2010*, pages 677–684, 2010.
- [4] J. A. Gallian. A dynamic survey of graph labeling (25th edition). *Electronic Journal of Combinatorics*, 5:Dynamic Survey 6, 43, 2022.
- [5] A. Godinho and T. Singh. Some distance magic graphs. *AKCE International Journal of Graphs and Combinatorics*, 15(1):1–6, 2018. [10.1016/j.akcej.2018.02.004](https://doi.org/10.1016/j.akcej.2018.02.004).
- [6] M. Jinnah. On  $\Sigma$ -labelled graphs. In B. Acharya and S. Hedge, editors, *Technical Proceedings of Group Discussion on Graph Labeling Problems*, pages 71–77. NITK Surathkal, 1999.

- 
- [7] N. Kamatchi. *Distance Magic and Distance Antimagic Labelings of Graphs*. PhD Thesis, Kalasalingam Academy of Research and Education, Krishnankoil, Srivilliputhur, Tamil Nadu, June 2012.
- [8] P. Kovář, D. Fronček, and T. Kovářová. A note on 4-regular distance magic graphs. *The Australasian Journal of Combinatorics*, 54:127–132, 2012.
- [9] M. Miller, C. Rodger, and R. Simanjuntak. Distance magic labelings of graphs. *The Australasian Journal of Combinatorics*, 28:305–315, 2003.
- [10] A. O’Neal and P. J. Slater. An introduction to distance  $D$  magic graphs. *Journal of the Indonesian Mathematical Society*, (Special edition):91–107, 2011.
- [11] A. O’Neal and P. J. Slater. Uniqueness of vertex magic constants. *SIAM Journal on Discrete Mathematics*, 27(2):708–716, 2013. [10.1137/110834421](https://doi.org/10.1137/110834421).
- [12] R. Pawar and T. Singh. Distance magic labeling of generalised mycielskian graphs. *arXiv:2306.07578*, 2023.
- [13] S. Rao, T. Singh, and V. Parmeswaran. Some sigma labelled graphs i. In S. Arumugam, B. Acharya, and S. Rao, editors, *Graphs, Combinatorics, Algorithms and Applications*, pages 135–140. Narosa Publishing House, New Delhi, 2008.
- [14] V. Vilfred.  $\Sigma$ -Labelled Graphs and Circulant Graphs. Ph.D. Thesis, University of Kerala, Trivandrum, Kerala, India, 1994.
- [15] D. B. West. *Introduction to Graph Theory*. Prentice Hall, 2nd edition, Sept. 2000.
- [16] F. Yasin and R. Simanjuntak. A heuristic for distance magic labeling. 74:100–104, 2015. <https://doi.org/10.1016/j.procs.2015.12.083>. “The 2nd International Conference of Graph Theory and Information Security”.

## Appendix

---

### Algorithm Distance magic graph generator

---

```

1: • To generate k-sum subsets
2: function GETSUBSET( $A, k, n, ans, List$ )
3:    $A \leftarrow$  list of integers from 1 to  $n$ 
4:    $ans \leftarrow$  empty list
5:    $List \leftarrow$  empty list
6:   if  $k = 0$  then
7:      $ans1 \leftarrow ans$ 
8:      $List.append(ans1)$ 
9:     return
10:  if  $k \neq 0$  and  $n = \text{length of } A - 1$  then
11:    return
12:  if  $k - A[n] \geq 0$  then
13:     $ans.append(A[n])$ 
14:     $getSubset(A, k - A[n], n - 1, ans, List)$  ▷ include the element
15:     $ans.pop()$ 
16:   $getSubset(A, k, n - 1, ans, List)$  ▷ do not include the element
17: • To generate neighborhood sets NS
18: function GENERATE_NEIGHBORS( $n, k$ )
19:   $getSubset(A, k, n - 1, ans, List)$ 
20:   $neighborhoods \leftarrow$  empty list
21:  for  $i \leftarrow 1$  to  $n$  do
22:     $N_i \leftarrow$  empty list
23:    for each subset in List do
24:      if  $i \notin$  subset then
25:         $N_i \leftarrow$  append subset to  $N_i$ 
26:    append  $N_i$  to neighborhoods
27:  return neighborhoods
28:  $adj \leftarrow$  list of empty lists of size  $n + 1$ 
29:  $sum\_list \leftarrow$  list of  $n + 1$  elements initialized to 0
30:  $cnt\_depth \leftarrow 0$ 
31:  $successful \leftarrow$  list containing a single element 0
32:  $N \leftarrow generate\_neighbors(n, k)$ 
33:  $N.insert(0, [])$ 
34: function IS_SYMMETRIC( $adj, n, depth$ ) ▷ Checks the condition in (1) in Step 3'
35:  for  $i \leftarrow n$  to  $depth - 1$  do
36:    for each neighbor in  $adj[i]$  do
37:      if neighbor  $\leq$  depth then
38:        continue
39:      if  $i \notin adj[neighbor]$  then
40:        return False

```

---

(Continued)

---

```

1: function CHECK(adj_list, sum_list)                                ▷ Checks the condition given in (2) in Step 3'
2:   done ← 1
3:   for  $i \leftarrow 1$  to length of adj_list do
4:     if sum_list[ $i$ ]  $\neq k$  then
5:       done ← 0
6:       break
7: • To construct a tree  $T(n, k)$  and explore the successful branch
8: function EXPLORER( $N$ , adj, depth, breadth, sum_list, successful)
9:   adj[depth] ←  $N[\text{depth}][\text{breadth}]$                                 ▷ Fixes first element of  $NS(n)$  as root of  $T(n, k)$ 
10:  if depth = 1 then
11:    for  $i \leftarrow 1$  to length of adj[depth] do
12:      sum_list[adj[depth][ $i$ ]] ← sum_list[adj[depth][ $i$ ]] + depth
13:    if check(adj, sum_list) and is_symmetric(adj, n, depth) then
14:      successful[0] ← successful[0] + 1
15:      for  $i \leftarrow 1$  to length of adj[depth] do
16:        sum_list[adj[depth][ $i$ ]] ← sum_list[adj[depth][ $i$ ]] - depth
17:      Print the adjacency list of the graph
18:      return True
19:      for  $i \leftarrow 1$  to length of adj[depth] do
20:        sum_list[adj[depth][ $i$ ]] ← sum_list[adj[depth][ $i$ ]] - depth
21:      return False
22:  flag ← is_symmetric(adj, n, depth)
23:  for  $i \leftarrow 1$  to length of adj[depth] do
24:    sum_list[adj[depth][ $i$ ]] ← sum_list[adj[depth][ $i$ ]] + depth
25:    if sum_list[adj[depth][ $i$ ]]  $> k$  then
26:      flag ← 0
27:  if flag = 1 then
28:    for  $i \leftarrow 1$  to length of  $N[\text{depth} - 1]$  do
29:      if explorer( $N$ , adj, depth - 1,  $i$ , sum_list, successful) then
30:        return True
31:  for  $i \leftarrow 1$  to length of adj[depth] do
32:    sum_list[adj[depth][ $i$ ]] ← sum_list[adj[depth][ $i$ ]] - depth
33:  adj[depth] ← empty list
34:  return False
35: for  $i \leftarrow 1$  to length of  $N[n]$  do                                ▷ construct  $T(n, k)$  rooted at each element in  $NS(n)$ 
36:  explorer( $N$ , adj,  $n$ ,  $i$ , sum_list, successful)

```

---