# Exhaustive Verification of Weak Reconstruction For Self Complementary Graphs

S. K. Gupta , Sahil Singla, Akash Khandelwal, Apurv Tiwari, Srilekha
{skg, cs1070181}@cse.iitd.ac.in, {akash3119, apurvtwr, srilekha.vk}@gmail.com

***Abstract*—This paper presents an exhaustive approach for verification of the weak reconstruction of Self Complementary Graphs upto $17$ vertices. It describes the general problem of the Reconstruction Conjecture, explaining the complexity involved in checking deck-isomorphism between two graphs. In order to improve the computation time, various pruning techniques have been employed to reduce the number of graph-isomorphism comparisons. These techniques offer great help in proceeding with a reconstructive approach. An analysis of the numbers involved is provided, along with the various limitations of this approach. A list enumerating the number of SC graphs up till $101$ vertices is also appended.**

## I. Introduction

### A. The Reconstruction Conjecture (RC)

*:* The Reconstruction Conjecture (RC) is one of the most celebrated unsolved problems in Discrete Mathematics and Combinatorics circles. It was first discovered by S.M. Ulam and P. J. Kelly in 1941 [3] . Any graph $G$ has a vertex set $V(G)$ and an edge set $E(G)$. A vertex-deleted-subgraph of $G$, $G_i$, is the unlabelled subgraph of $G$ with the $i^{th}$ vertex and its coincident edges removed. The deck of the graph $G$ is the collection of all vertex-deleted subgraphs of $G$. For terms not defined here, we shall use the terminology followed in Harary [13].

*1) Original Definition:*

*:* Ulam [27] states the following problem: "Suppose that in two sets $A$, $B$; each of $n$ elements, there is defined a distance function $\rho$ for every pair of distinct points, with values either $1$ or $2$ and $\rho(x, x) = 0$. Assume that for every subset of $n-1$ points of $A$; there exists an isometric system of $n-1$ points of $B$, and that the number of distinct subsets isometric to any given subset of $n-1$ points is same in $A$ as in $B$. Are $A$ and $B$ isometric?"

*2) Modified Definition of the Graph Reconstruction Conjecture:* Reconstruction Conjecture [13]: "A simple finite graph $G$ with at least three points can be reconstructed uniquely (up to isomorphism) from its collection of vertex deleted subgraphs $G_i$." This conjecture was called by Harary [13], a "graphical disease", along with the 4-Color Conjecture and the characterization of Hamiltonian graphs.

### B. Reconstructive Approach Towards RC

*:* The reconstruction problems provide a fascinating study of the structure of graphs. The identification of structure of a graph is the first step in its reconstruction. We can determine various invariants of a graph from its subgraphs, which in turn tell us about the structure of the graph.

*:* One of the ways for tackling the RC is known as the reconstructive approach, and is followed in many of the proofs of the conjecture for specific classes. This approach relies on two parts: one- "recognizability", and two "weak reconstructibility"[3]. The class of graphs $C$ to which $G$ belongs is said to be recognizable if every reconstruction of $G$ belongs to the class $C$. The class $C$ is said to be weakly reconstructible if every reconstruction of a graph $G$ belonging to the class $C$ is isomorphic to $G$, for each $G$ in $C$ [3].

*:* A parameter of $G$ which can be deduced from the deck is called reconstructible. Another approach attempted by many is in determining reconstructibility of a graph on the basis of their graph invariants. Various properties such as characteristic polynomial [9], degree sequence [28] and diameter [25] have been proven to be reconstructible for SC graphs. Additionally, graph invariants like number of vertices, edges, blocks, cut-vertices, independent cycles and connectivity have been proven to be reconstructible [16].

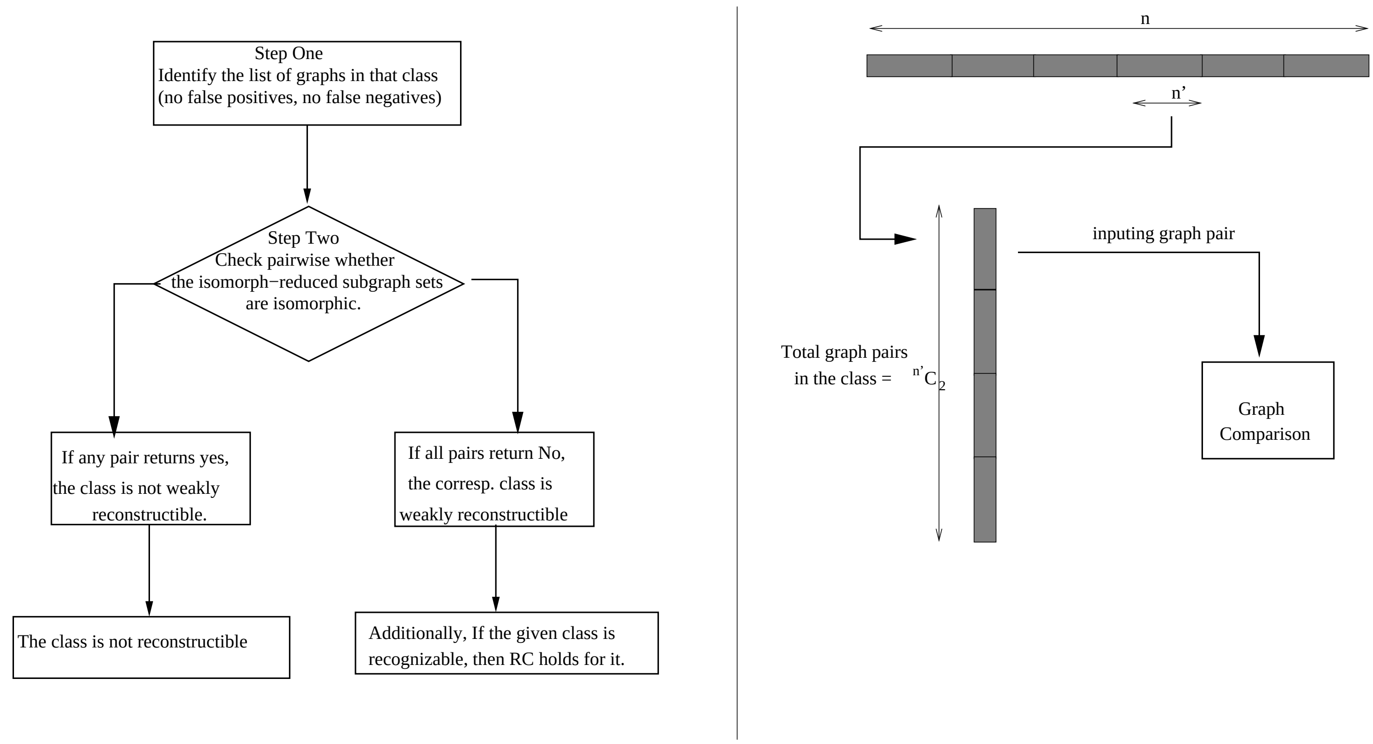


Fig. 1. The Procedure for RC Verification, depicted as (i)Overall Flowchart (ii)Line Diagram for the Approach Employed

### C. Discussion about the Problem

*:* McKay[1] employed an exhaustive technique to show that graphs up to $11$ vertices are determined uniquely by their sets of vertex-deleted subgraphs, even if the set of subgraphs is reduced by isomorphism type.

*:* The statement of the conjecture excludes the trivial graph $K1$, graphs on two vertices and infinite graphs. The deck of graphs on two points, i.e. $K2$ and $K2'$,are clearly homomorphic (a pair of $K1$s comprising each of their decks) but the graphs are non-isomorphic. For every infinte cardinal $\alpha$, there exists a graph with $\alpha$ edges which is not uniquely reconstructible from its family of edge deleted sub-graphs [7] . Apart from these two exceptions which prohibit the conjecture from encompassing all graphs, unique reconstructibility is conjectured for all other graphs.

*:* The conjecture has been proved for a number of infinitely sized classes of graphs, such as trees [15], squares of trees [24], unicyclic graphs [18], regular graphs [21] and disconnected graphs [14]. Though the problem can be stated very simply,yet due to a lack of a nice set of characterizing invariants, it has still not been proven for very important classes of graphs like bipartite graphs [3] and planar graphs [3]. For further study of this problem, the reader is referred to survey by Bondy [3].

*:* In a probabilistic sense, it has been shown that almost all graphs are reconstructible [2]. This means that the probability that a randomly chosen graph on $n$ vertices is not reconstructible goes to $0$ as $n$ goes to infinity. In fact, it was shown that not only are almost all graphs reconstructible, but also that the entire deck is not generally necessary to reconstruct them almost all graphs have the property that there exist three cards in their deck that uniquely determine the graph.

## II. RC Verification for Self-Complementary Graphs

### A. Overview of Procedure Employed

*:* The problem definition refers to unlabeled graphs making it computationally expensive(refer Sec.II-B for the worst case analysis). An idea of the numbers that are associated with the problem, is presented in Sec.II-D The appendix lists the variation in the number of SC graphs with the number of vertices.

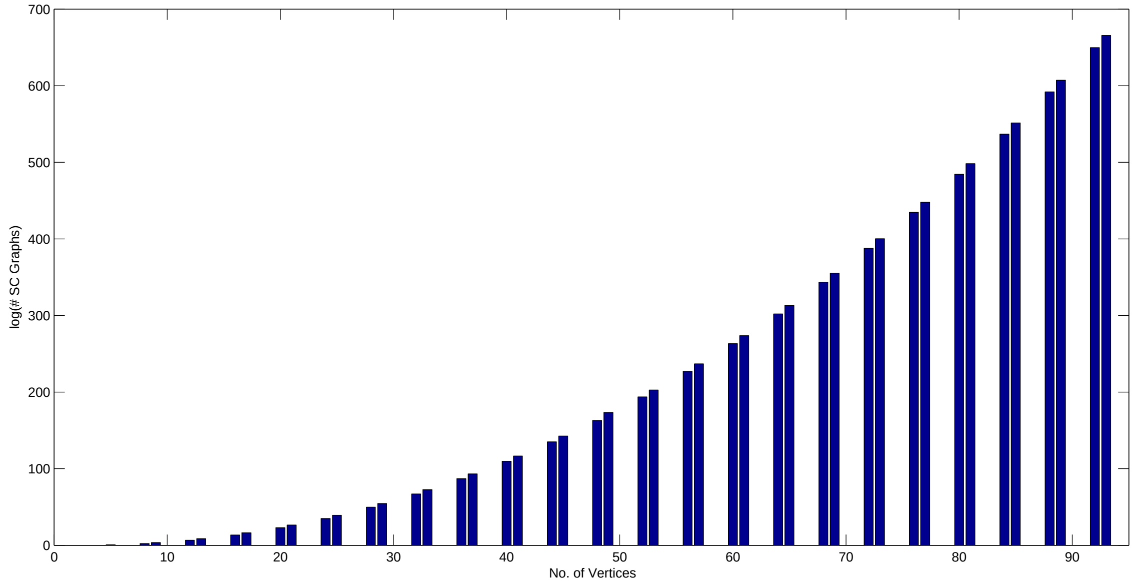


Fig. 2. A graph showing logarithm of No. of SC Graphs vs No. of vertices

*:* This paper suggests techniques through use of efficient structures for storage and suitable classifications(described in Sec.II-E). The intelligence of such techniques lies in reducing the complexity of the underlying problem.

*:* For detailed a survey of self complementary graphs, reader is referred to [11]. The listing of self-complementary graphs is available upto 17 vertices [6]. This listing of self complementary graphs is generated using the fact that every self complementary graph ($G$) on $4n$ vertices can be broken down into edge-disjoint subgraphs $H$, $H^*$ and $B$, such that $G = H + B + H^*$ , $H' = H^*$ and $B$ is a bipartite graph between the vertex sets of $H$ and $H^*$. There exists a self-complementing permuation (sigma) with even length cycles. This permutation plays a key role in generating the self complementary graphs [22]. An efficient way of generating SC graphs on $4n+1$ vertices using the set of SC graphs on $4n$ vertices is discussed in [29]. Although this process is exhaustive, it creates multiple copies of a graph in the form of isomorphic graphs. In [19] , a method is suggested to reduce the generation of such copies of graphs. In this procedure, the permutations of vertices within a cycle of a self-complementing permutation are avoided as each such permutation generates the same set of graphs.

*:* In proving that self complementary graphs up to 17 vertices are uniquely determined (within the set of all graphs) by their decks, the algorithmic challenge lies in reducing the number of comparisons among graph-pairs. The flowchart of the procedure adopted is in Fig. 1. The number of cases for isomorphism checking were reduced by a large extent by obviating inter-class comparison through classification [Sec. II-Ca].

*:* Within each class, isomorphism among unlabeled decks had to be checked to see if they can uniquely identify a graph for all possible pairs of graphs, which constituted the most frequent step. Although such comparisons were reduced by a large number through classification of graphs, it still is the major contributor to the execution time in this module. To avoid this, the graphs in the decks are again classified on the basis of degree sequence. (Refer Sec.II-Cb for details). The deck-isomorphism checking was done based on the structure represented in Sec. II-E.

*:* At the lowest level of classification, graphs required for comparison were taken pairwise; hence dreadnaut interface to nauty [5] was used, for individual isomorphism-checking. The details of the

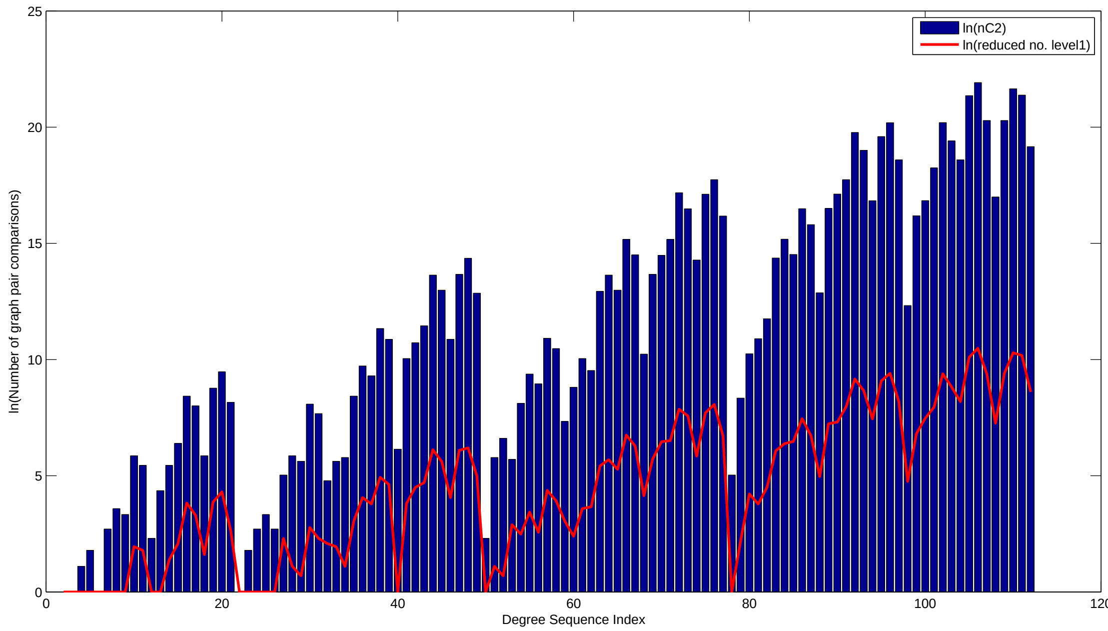


Fig. 3. A graph showing reduction in the number of comparisons after Level-1 Pruning

structure used for this has been discussed in Sec. II-E. The choice of Brendan McKays Nauty(No AUTomorphisms, Yes?)[B5] for use in our exhaustive verification is based on the experimental survey of Graph Isomorphism Algorithms in [10] which gives a comprehensive assessment of various GI algorithms which implement exact one-to-one matching using various techniques. Nauty was found to have the best performance time for small moderately dense graphs.

### B. *Complexity Issues*

*:* For any class of graphs, containing say $n$ graphs, with $m$ vertices, the procedure requires comparing every pair (${}^{n}C_2$ in this case) for deck-level isomorphism, which in turn requires a worst case of ${}^{m+1}C_2$ isomorphism checks. In any iteration, two graphs are read from the file, and their decks are kept in main memory, each deck having $m$ graphs, stored as adjacency matrices.

*:* Time Complexity(Worst Case):

$${}^{n}C_2 * \ {}^{m+1}C_2 * O\{GI^1\}$$

[1] The graph isomorphism problem is one of a very small number of problems belonging to NP neither known to be solvable in polynomial time nor NP-complete [12], and a special complexity class $GI$ has been defined for such problems.

### C. *Pruning*

*:* As discussed in the previous section, the complexity involved in checking RC for a certain class is very high as it involves the comparison of the unlabeled decks of graph pairs. So, in order that lesser number of such comparisons are performed, the graphs should be partitioned into mutually exclusive and exhaustive classes, so that inter-class graph pairs need no isomorphism checking. In our approach, pruning was employed at two levels, where various parameters like degree sequence, characteristic polynomial, diameter were used to prune the graphs.

*Level-1 :* The listings of SC graphs were first classified according to degree sequences, so that any two graphs with different degree sequences dont need to be compared. The divisions formed on basis of degree sequence were further classified into groups of graphs with the same characteristic polynomial. Two graphs in the same group were compared only if their diameter was same.

Let the set of all classes thus formed be $S$, then the number of graph comparisons reduces as

$${}^{n}C_2 \rightarrow \sum_{i\epsilon[\#S]} {}^{n_i}C_2$$

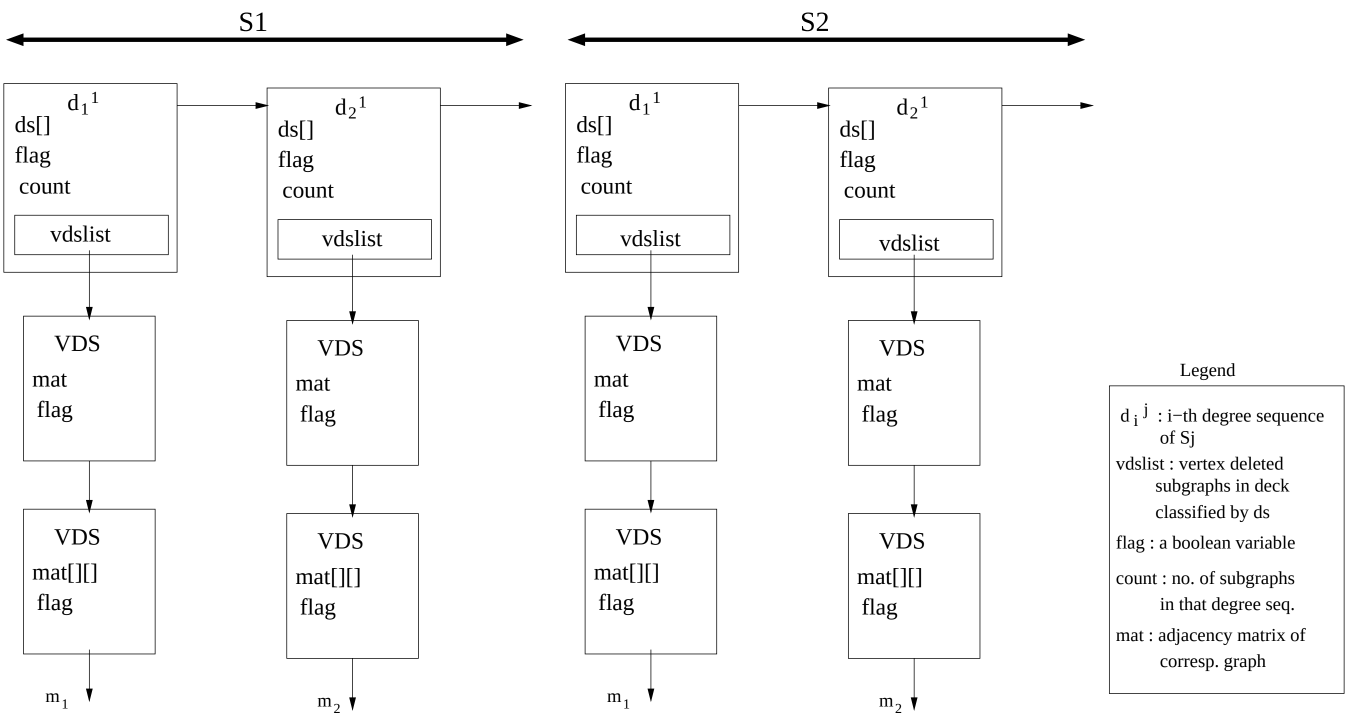


Fig. 4. The data structures used while comparing decks for isomorphism

Fig. 3 depicts the improvements observed when all SC graphs over 16 graphs are pruned. The bars correspond to different classes with heights representing the no. of pairwise comparisons required within that class. These pairs are passed on to the Graph Comparison module. The continuous curve intersecting the bars represents the actual number of pairs compared.

*Level-2:* This level reduces number of isomorphism checks required in a graph comparison between some graphs $G_1$ and $G_2$. The graphs in corresponding decks $D1$ and $D2$ are classified according to degree sequences.

The deck isomorphism is checked only if the number of graphs of any degree sequence in both the decks are equal.

Let $S_i$ be the set of degree sequences of the graphs in $D_i$. Then the number of graph isomorphism checkings reduces as:

$${}^{m+1}C_2 \rightarrow \sum_{i\epsilon[S_1 \cap S_2]} {}^{m_i+1}C_2$$

## D. Analyzing the Numbers Involved

*:* As is clear from the plot shown [Fig. 2], the number of self complementary graphs rise steeply with the increase in the number of vertices. Thus, as one proceeds further, the storage as well as computation time becomes a dominant factor while analyzing the graphs. A detailed formula for the numbers of SC graphs can be found in the appendix.

*:* Additionally, as the number of vertices increase, the size of individual graph also increases. (equivalently the number of ascii characters required to encode an individual graph in graph6[3] format).

*:* Both these factors clearly indicate the sharp increase in the problem complexity on increasing the number of vertices. The approach employed to deal with both space and time factors have been have been discussed in further sections.

## E. Storage and Implementation details

*:* The graphs were encoded into the graph6 format and stored as character strings in files. While comparing two graphs, the corresponding decks were stored in a structure as shown in Fig. 4, to speed-up the deck-isomorphism checking. In case the two decks were concluded to be different at any stage, the next graph pair was considered. If a pair of decks are found to be isomorphic, it can be concluded that the class under consideration is not uniquely reconstructible.

*:* Each graph's deck has been stored as a list of its degree sequences comprising the vertex-deleted graphs. When the vertex-deleted subgraphs are being formed from the main graph, their degree sequences are calculated and they are appended to the existing structure. The number of subgraphs belonging to each degree sequence is stored as 'count', which is used as a measure of potential dissimilarity between the decks under inspection.

*:* The various terms used have been explained below:

- $VDS$ (Vertex Deleted Subgraphs): to store a vertex deleted subgraph of the current graph i.e. one graph of the deck, along with its degree sequence.
- $DS$ (Degree Sequence): to store together all the vertex deleted subgraphs of a graph that have the same degree sequence, and their count.
- $mat[][]$: represents the adjacency matrix of the graph.
- $ds[]$: is the degree sequence array for the graph.

## III. Results, Limiting Factors and Future Work

*:* An exhaustive approach was followed in order to work towards disproving the conjecture for Self Complementary graphs. Since no counter example was found up to graphs on 17 points, the weak reconstruction was established for all SC graphs up to 17 vertices.

*:* The approach is limited by various factors. The number of SC graphs increases more than exponentially and generation and storage of these graphs becomes a problem. If we go by the way of exhaustive checking of RC for the whole class, the deck isomorphism checking for unlabeled graphs involves a large amount of computation(As discussed in Sec.II-B , the isomorphism problem is neither Polynomial time, nor NP-Complete). Thus, progress using exhaustive approach is limited by the computational power.

*:* A possible approach could be to prune graphson the basis of various properties to form classes, accounting for the rapid increase in numbers(Sec.II-D). These classes are not necessarily mutually disjoint but jointly exhaustive, such that inter-class graph comparisons are not necessary, thereby reducing the amount of computation required, in terms of both space and time. Cluster computing or parallel programming can be used, but that can take the endeavor only one step further. A search for a counterexample can end only when one has actually been found.

*:* Having validated the weak reconstruction of Self-Complementary graphs through exhaustive verification up to 17 vertices, proving the Reconstruction Conjecture for SC Graphs requires the establishment of their Recognizability.

## References


[1] McKay, B.D. *Small graphs are reconstructible,* Australasian. J. Combin.,15(1997), 123 - 126.

[2] Bollobas, B. *Almost every graph has reconstruction number three,* J. Graph Theory,14(1990), 1-4.

[3] J. A. Bondy & R. L. Hemmingerpg , *Graph Reconstruction-A Survey,* J. Graph Theory 3(1977), pp227

[4] McKay, B.D., *Details about graph6 format used,* [http://cs.anu.edu.au/∼bdm/data/formats.html] (as on 14-10-09)

[5] McKay, B.D., *The nauty web page,* [http://cs.anu.edu.au/∼bdm/nauty] (version 2.2 as on 14-10-09)

[6] McKay, B.D., *Combinatorial Data Page* [http://cs.anu.edu.au/∼bdm/data/graphs.html]

[7] Carsten Thomassen, *Counterexamples to edge reconstruction conjecture for infinite graphs*, Discrete Mathematics, Vol. 19, Issue 3, (1977) pages 293-295.

[8] Dieter Kraiscli & Lane A. Hemachandra, *On the Complexity of Graph Reconstruction,* March 11, 1991

[9] Elias M. Hagos, *The characteristic polynomial of a graph is reconstructible from the characteristic polynomials of its vertex-deleted subgraphs and their complements,* Electr. J. Comb. 7:(2000)

[10] P. Foggia, C.Sansone, M. Vento, *A Performance Comparison of Five Algorithms for Graph Isomorphism,* proceedings of 3rd IAPR TC-15 Workshop on Graph-based Representations in Pattern Recognition, (2001).

[11] Alastair Farrugia, *Self-complementary graphs and generalisations: a comprehensive reference manual,* University of Malta, August 1999

[12] Garey, Michael R.; Johnson, David S., *Computers and Intractability: A Guide to the Theory of NP-Completeness,* W. H. Freeman, ISBN 978-0716710455, OCLC 11745039

[13] F. Harary, *Graph Theory,* (2001), Tenth re-print Narosa publishing house.

[14] F. Harary, *On the reconstruction of a graph from a collection of subgraphs,* In Theory of Graphs and its Applications, pages 47-52. Prague, 1964.

[15] P.J. Kelly, *A congruence theorem for trees,* Pacific J. Math., 7:961-968, 1957.

[16] A.D. Keedwell, *Surveys in Combinatorics, 1991,* Cambridge University Press.

[17] Manvel, B. & Reyner, S.W., *Subgraph Equivalence of Graphs,* J. Comb. Inf. And Syst. Sci. 2(1976), pp. 41-47

[18] B. Manvel, *Reconstruction of unicyclic graphs. In Proof Techniques in Graph Theory,* pages 103-107. Academic Press, New York, 1969.

[19] Myles F. McNally & Robert R. Molina, *Cataloging Self-Complementary Graphs of Order Thirteen,* Congressus Numerantium Vol. 108, 11-15 (1995)

[20] C. St. J.A. Nash-Williams, *Reconstruction of infinite graphs,* Discrete Mathematics Vol. 95 (1991) 221-229.

[21] C. St. J.A. Nash-Williams, *The reconstruction problem,* in Selected Topics in Graph Theory, pages 205-236. Academic Press, 1978.

[22] P. S. Nair, *Note Construction of self-complementary graphs,* in Discrete Mathematics Vol. 175 (1997) 283-287.

[23] R.C. Read, *On the number of self-complementary graphs and digraphs,* J. London Math. Soc. 38 (1963) 99-104

[24] Gupta S. K, *Reconstruction conjecture for square of a tree,* Volume 1073/1984, Springer Berlin / Heidelberg, Graph Theory Singapore 1983, pages 268-278.

[25] S.K. Gupta, Pankaj Mangal & Vineet Paliwal, *Some work towards the proof of the Reconstruction Conjecture* , Discrete Mathematics Vol. 272 (2003) pages 291-296.

[26] Sloane, N. J. A., *The On-Line Encyclopedia of Integer Sequences,* Sequence A000171/M0014 [http://www.research.att.com/∼njas/sequences/Seis.html]

[27] Ulam S.M., *A collection of Mathematical Problems* (Wiley-Interscience), New York(1960), pp. 29

[28] Wendy Myrvold, *For all graphs G on 7 or more vertices, the degree sequence can be reconstructed from any n-1 subset of cards where n is the number of vertices of G,* Discrete Mathematics, Vol. 102, Issue 2, May 1992, pages 187-196.

[29] Jin Xu, C.K. Wong, *Self-complementary graphs and Ramsey numbers Part I: the decomposition and construction of self-complementary graphs*, Discrete Mathematics, Vol. 223 (2000), pages 309-326.


## APPENDIX

*:* For any natural number $n$, there are no SC graphs on $4n+2$ and $4n+3$ vertices, [23]. Listing of no. of SC graphs up till 31 vertices is given by Sloane[26]. Table at the bottom gives the listing up till 101 vertices. It has been computed using the following formula given by [23].

*:* Let $\sigma_k$ be the number of SC graphs on k vertices, $d(q,r)$ be the highest common factor of $q$ and $r$, $(j)$ denotes summation for $j_1+2j_2+3j_3+...+nj_n=n$ and $k_s=j_{4s}$, then

$$\sigma_{4N} = \sum_{(k)} \frac{2^R}{\prod_{s=1}^{N} s^{k_s}.k_s!}$$

where

$$R = 2\sum_{s=1}^{N} k_s(sk_s-1)+4\sum_{1\leq\alpha<\beta\leq N} k_\alpha k_\beta d(\alpha,\beta)$$

and

$$\sigma_{4N+1} = \sum_{(k)} \frac{2^{R'}}{\prod_{s=1}^{N} s^{k_s}.k_s!}$$

where

$$R' = \sum_{s=1}^{N} k_s(2sk_s-1)+4\sum_{1\leq\alpha<\beta\leq N} k_\alpha k_\beta d(\alpha,\beta)$$

| No. of vertices | No. of SC Graphs | No. of digits |
|---|---|---|
| 1 | 1 | 1 |
| 4 | 1 | 1 |
| 5 | 2 | 1 |
| 8 | 10 | 2 |
| 9 | 36 | 2 |
| 12 | 720 | 3 |
| 13 | 5600 | 4 |
| 16 | 703760 | 6 |
| 17 | 11220000 | 8 |
| 20 | 9168331776 | 10 |
| 21 | 293293716992 | 12 |
| 24 | 1601371799340544 | 16 |
| 25 | 102484848265030656 | 18 |
| 28 | 3837878966366932639744 | 22 |
| 29 | 491247277315343649710080 | 24 |
| 32 | 128777257564337108286016980992 | 30 |
| 33 | 32966971058719932671168222859264 | 32 |
| 36 | 6145487749730846261818853233041 0573824 | 38 |
| 37 | 3146489675114846976177661243674 1418123264 | 41 |
| 40 | 4223146893959501354337304999580 70655419345928192 | 48 |
| 41 | 4324502413750846252038423855257 12986695638650716160 | 51 |
| 44 | 4221271913164542277754835626477 904283804666087301941 5068672 | 59 |
| 45 | 8645164877296082081022797352234 46130127050895825107423833620 48 | 62 |
| 48 | 6188450068892835670911232879663 24958874629599640950150338415 26613475328 | 71 |
| 49 | 2053478914819948612803903676075 18623249435827483168144028565 4319521071104000 | 76 |
| 52 | 1339918336786507094228440210984 22189855201413567485644125926 0891574069074668524929024 | 85 |
| 53 | 1097661102231728262196282094323 29898030465321728087071079183 821320575674606661239801118 72 | 89 |
| 56 | 4310325518789747596337196011287 33652032040321118483003619093 69371349100032415299893295782 3483510784 | 99 |
| 57 | 7062037329984837069952278475963 08048394508719529822381377043 04544595852262442019899568412 38095158312960 | 103 |
| 60 | 2070611242694996407173747823074 15600562906919880543389280838 91796613389787117940417704898 211482867551464828017049 60 | 115 |
| 61 | 678497892006294445422241289323780963772966250550903395594719662197061934532684847875764204065011307980559179643900 39552 | 119 |
| 64 | 149203264336484720095016247814949538021562046303278771936611682711188585667279315272321413811185196875595690865500457245516599656448 | 132 |
| 65 | 977818513155586058065330879779815922582228413241077407349841104091685122324386083797116636681567831410799875295740469800321300678102 2208 | 136 |
| 68 | 161900848951598125922102268348274406927651321450449665581675617732454860248879488944014897508474614218151466763396272067235453567049272027118181548032 | 150 |
| 69 | 21220668073783869247959864307156165484013171654650772916763943394029199297540341718175286846699876125635003159535688628862636060376108647802659092677263360 | 155 |
| 72 | 2654705356379055929204863712989313218127639031800769404053274594403808493604137351010498386075022562444691478778034248873731700346613195686045490049111661980510574346240 | 169 |
| 73 | 695915080942631236783977078669218752668281018309497989765767218998018679614933199959215730269451918584309173295516981430082962221077114245985284385770379081577130059748081664 | 174 |
| 76 | 659815347203130704718396676527918606269829296456485444576342226280280781022061748227086634768047400194662842649127632935670482530781830425344046435419624192224493419861363519618091086512128 | 189 |
| 77 | 345933268754420835150345813829534665185861078612082569791333115810000083686448692744624238401839138403084821404189180884505032830131842815959426804814410030249788029524831615029411122045482172416 | 195 |
| 80 | 24927119044119477117844422421801142750406318763668552125446880650874407762280575197194575838977943780581367051855024858108280083870738786320868611120029970883813893542542944411254201506056420775390020658580684 80 | 211 |

| No. of vertices | No. of SC Graphs | No. of digits |
|---|---|---|
| 81 | 2613797877880662483818937844323587485999371856337376557313284712987112359750241420077874674988206302205249585377958004134277740830696380612038122291457789670024121518754058314562070614073783848539402108466067274727424 | 217 |
| 84 | 143500180100207479114997189998243862065540710190653850003333819219400329763364736982700570090472547380911009106852674830812739206191280315369399095478437471175257155465483947177581332000048787390822621023311840029743733810242498068480 | 234 |
| 85 | 300941689697510315240870035084081885319109136917752219573706986858476981024828585725084639468252754069591552848951249304303042586471904134865102515972545201114567895301002357392115989584048323232069154568737481977621471324827215676259172352 | 240 |
| 88 | 126168052976356528017519464279970380277246821645939532372117786703617869491956079915156082774256576345576135739799953970896301446268644234425118898688586793035588197774593492959263752273662468394418895918966369058683145825983584770262848456903118303090704384 | 258 |
| 89 | 529187169270944090889981319556369590445137244678795761778453210660885257808496158528337839545292996698799615169820977453514666532275140114735433351477361281588332949207084604669740372448219564012348196733416859665668093100517163412258909324155360897097785281609728 | 264 |
| 92 | 1697700918022942826236799806104229489874803275085747485902192510316878190849306525453360814303037110889858541184777857583080786967028277756375448174045869211032708723308858026610776013037158540989021523348251443911867845272206923275150487068038869929119817086324226309465545826631680 | 283 |
| 93 | 14241347502534602375712605463648940411059694580953496472176059408999706473402929248747922469461962203935199750224295087028364360360907798939131679386970250620149300471441725739950716781658847638358034833426351264367667054831231390450606607821527537806161643855011241390200362471977318875136 | 290 |
| 96 | 350275323525357999981371792335876318434320809029842859164255150334412339862643411847434226169395893161240502077884123858229470450551749606322076202245751917385494313364616822116714785536672732266008431592758979882511845456028102786108100130928084285002917195332436745689459703629357363973516979987822046871552 | 309 |
| 97 | 5876644762254812643015469881224156427065672743585987058020545333727681840971454787671682854617943027039896164325366473176779038921323940812052568930906975634286942352175422358734001443209038841858865191992332046723165454934187325296953359420971823490910999557582324280710649644811822339985293109085215730542026162176 | 316 |
| 100 | 1110066810280143278346105974280473566442343518932753410230043852951741884650937161269266154280683695138791888369309588232985338008853156894716218336870085824373849823607818018769274953434300234421558703014674009040015882673946655163809537590676885893155260788167350906508845913270714972917156241354152457619380511504819821410166163636224 | 337 |
| 101 | 37247661301001968583521485096708950624756090661926566993955723864558299168132980980190826546028945355564840546206129732188699196088322938732531579258238351046911136773564124827301206921349708331261088926445867614264528750147164300653121910908203986135954144070339910422449994155010670066943476494184925439729103104685100083374663199397546819584 | 344 |