P2.10.2 High degree orbit-graph can be isomorphic or coincide with their low degree basic graph.

P2.10.3. A high degree orbit structure can be *reconstruct its basic structure*, if the last is bi- or mono-symmetric.

P2.10.4. Inducing the high degree orbit-structures is a *convergent process*, it close with crop up a low degree graph.

So far we treat orbit structures of vertex symmetric graphs. It has thought to treat also "sign graphs" of *partially*- and *0-symmetric* graphs.

Proposition 2.11. A graph, whose edges e_{ij} correspond to a certain class pair signs ***dnq*** of ***G*** is a ***sign graph*** G_{dnq} of ***G.***.

Comment: The *sign graphs* G_{dnq} can be open the "hidden sides" of partially- and *0*-symmetric graphs.

Symmary The main matter of orbit-structures consist in recognition the hidden properties of a basic structure. By help orbit-structures can be find the same attributes of various structures. For example, Hypercube and Möbius-Kantor graph have some common orbit-structures, an orbit-structure of Folkman graph is Petersen graph, etc.

3. PROBLEMS OF CANONICAL REPRESENTATION AND ISOMORFISM

3.1. Canonical representation the graphs

Canonical representation of a graph means it show in a form which represent its *structure*, recommendatory *with exactness up to isomorphism*. This problem set up presumably Lazlo Babai [11, 12] in year 1977. Canonical representation constitute, for example *3-cube codes* [13]. Unfortunately the no contain data about the structure self.

More deeply can be invade by semiotic invariants.

Proposition 3.1. Sign matrix ***S*** is a ***canonical representation*** of graph with exactness up to its *structure,* i.e. up to *pair signs, orbits* and *isomorphism***.**

Thus, such canonical representation is more than isomorphism recognition. It is recognition of the structure completely, where essential role has orbit recognition.

Example 3.1. A canonical presentation of a *partially symmetric and strongly regular graph* ***WEI*** [14, p 166(1)], its decomposed by ***u***- and ***s***-vectors sign matrix ***S*** and structural properties:

```
A:-2.8.20; B:-2.8.19; C:-2.8.18;
```
`D:+2.7.13; E:+2.7.14; F:+2.7.15.`

```
|1  1| 2  2| 3 3| 4  4| 5| 6  6| 7  7| 8 8| 9  9|10 10|11|12|13|14 14|15|        u_i orb
20 24|12 14| 1 2| 9 19| 6|10 16| 8 18| 4 7|11 17|13 15|23| 3|22|21 25| 5|  i  ABCDEF   *
 0  F| C  C| C B| F  C| C| B  F| C  E| F C| E  F| E  C| B| F| F| F  C| F| 20  039039   1
    0| C  C| B C| C  F| C| F  B| E  C| C F| F  E| C  E| B| F| F| C  F| F| 24  039039   1
     | 0  F| F C| C  C| B| F  C| B  F| B F| E  C| F  E| F| C| F| F  C| E| 12  039039   2
          0| C F| C  C| B| C  F| F  B| F B| C  E| E  F| F| C| F| C  F| E| 14  039039   2
           | 0 F| F  C| E| F  C| F  B| F E| F  C| F  C| C| F| B| C  C| E|  1  039039   3
               0| C  F| E| C  F| B  F| E F| C  F| C  F| C| F| B| C  C| E|  2  039039   3
                | 0  F| E| F  C| F  E| F C| B  F| F  C| F| B| E| C  B| C|  9  039039   4
                     0| E| C  F| E  F| C F| F  B| C  F| F| B| E| B  C| C| 19  039039   4
                      | 0| C  C| B  B| B B| E  E| F  F| F| F| C| F  F| C|  6  066066   5
                         | 0  B| F  E| F E| B  E| B  E| B| B| C| E  E| C| 10  066066   6
                              0| E  F| E F| E  B| E  B| B| B| C| E  E| C| 16  066066   6
                               | 0  C| F B| F  B| C  C| E| C| B| C  E| E|  8  066066   7
                                    0| B F| B  F| C  C| E| C| B| E  C| E| 18  066066   7
                                     | 0 C| C  C| B  E| C| E| E| E  B| B|  4  066066   8
                                         0| C  C| E  B| C| E| E| B  E| B|  7  066066   8
                                          | 0  B| E  F| C| B| B| E  C| F| 11  066066   9
                                               0| F  E| C| B| B| C  E| F| 17  066066   9
                                                | 0  B| B| C| E| B  F| B| 13  066066  10
                                                     0| B| C| E| F  B| B| 15  066066  10
                                                      | 0| E| D| F  F| E| 23  066147  11
                                                         | 0| E| F  F| D|  3  066147  12
                                                            | 0| B  B| B| 22  093174  13
                                                               | 0  E| A| 21  147066  14
                                                                    0| A| 25  147066  14
                                                                     | 0|  5  255174  15
```

General invariants and measures (values):

Symmetry	**|V|**	**|R|**	***K***	***N***	***SVV***	***SV***	***SRV***	***HR***	***SR***
Partial symmetry	**25**	**300**	**15**	**154**	$1^{5}2^{10}$	*0.1723*	$1^{20}2^{128}4^{6}$	*2.1576*	***0.1290***

Specified invariants and measures of ***WEI*** and its complement ***WEIC***:

G	**|E|**	***k***	N^{+}	N^{-}	***P***	***CL***	***MC***	***DM***	SEV^{+}	***SE***	***TRA***	***BRA***
WEI	150	1	80	74	6	4	3	2	$1^{12}2^{67}4^{1}$	*0.1310*	*1.000*	*0*
WEIC	150	1	74	80	6	4	3	2	$1^{8}2^{61}4^{5}$	*0.1494*	*1.000*	*0*

Comments:

a) On the ground of only ***six pair signs*** is the 25×25 sign matrix decomposed by help *u*- and *s*-vectors to ***15 vertex orbits*** and ***115 partial matrices*** $W_{ki,kj}$.

b) 150 “non edges” of ***WEI*** form ***74 pair(–)orbits***, where –*A* form an orbit with two elements, –*B* form 33 orbits, among these 4 with one element and 29 with two elements, –*C* form 40 orbits, 4 with one, 31 with two and 5 with four elements.

c) 150 edges of ***WEI*** form ***80 pair(+)orbits***, where +***D*** 2 with one element, +***E*** 32 orbits, among these 4 with one, 27 with two and one with four elements, and +***F*** form 46 orbits, 6 with one and 40 with two elements.

d) B. Weissfeiler [14] a much not, who find that vertex orbits are essential attributes of graph structure. To pair orbits he but yet no subside. He had designed some strongly regular graphs that be grounded on the same pair signs, but are no isomorphic. On structural aspect: these differ from decompositions.

e) Graph ***WEI*** and its complement ***WEIC*** is ***triangular, 2-distance-*** and ***12-degree regular.*** Complement ***WEIC*** is also ***strongly regular.***

NB! For analyzing the structure of ***WEI*** is suitable use its *sign graphs* $WEI_{+2.7.13}$, $WEI_{+2.7.14}$ and $WEI_{+2.7.15}$.

3.2. Deep identification the structure

In the case of large vertex symmetric graphs can be the pair signs ±$\boldsymbol{d.n.q.}_{ij}$ stay insufficient to for differentiation the pair orbits $\boldsymbol{\Omega R}_n$. There we use *deep identification.*

<u>Propositions 3.1.</u> The methods of ***deep identification***:

<u>P3.1.1.</u> Using a *high degree* ***m*** pair graphs $\boldsymbol{g}_{ij}^{\boldsymbol{m}}$. A second degree pair graph $\boldsymbol{g}_{ij}^{\boldsymbol{m}}$ arise between vertices ***i*** and ***j*** of a graph ***G*** when to remove from this a "kernel" of pair graph $\boldsymbol{g}_{ij}$, i.e. $\boldsymbol{g}_{ij}^{m=2}=\boldsymbol{G}\backslash[\boldsymbol{g}_{ij}\backslash(\boldsymbol{v}_i\&\boldsymbol{v}_j)]$.

<u>Comment:</u> In such case is to *complementary identifier* the pair sign $\boldsymbol{dnq}_{ij}^{\boldsymbol{m}}$ of its high degree pair graph $\boldsymbol{g}_{ij}^{\boldsymbol{m}}$. Such deep identification we call ***high identification.***

<u>P3.1.2.</u> Using a *local sign matrix* $\boldsymbol{S}_{ij}$ for identify the first or high degree pair graphs.

<u>Comment:</u> In such case take the *complementary identification* place on the ground local sign matrices $\boldsymbol{S}_{ij}$. Such deep identification we call ***local identification.***

<u>P3.1.3.</u> Using a *product* $\boldsymbol{E}\times\boldsymbol{E}\times\boldsymbol{E}\times\ldots=\boldsymbol{E}^{\boldsymbol{n}}$ of an adjacency matrix, whereof elements $\boldsymbol{e}_{ij}^{\boldsymbol{n}}$ identify the *most length paths* between vertices.

<u>Comments:</u> Thus, it is touch anew with a *pair graph* where its invariants $\boldsymbol{e}_{ij}^{\boldsymbol{n}}$ constitute also pair signs, but these no contain structural data. Such deep identification we call ***product identification.***

The product identification is used more.

<u>Product Identification Principle.</u> OPERAND: *List of adjacent elements* ***L.*** ALGORITHM: 1) Form the adjacency matrix ***E.*** 2) Multiple it with itself $\boldsymbol{E}\times\boldsymbol{E}\times\boldsymbol{E}\times\ldots=\boldsymbol{E}^{\boldsymbol{n}}$ and fix in case of each degree ***n*** the number ***p*** of obtained differences among *productive pair signs* $\boldsymbol{e}^{\boldsymbol{n}}_{ij}$, which as rule enlarge. 3) If ***p*** more no enlarge, then to stop the multiplication and to fix the last product $\boldsymbol{E}^{\boldsymbol{n}}$, corresponding ***p*** and successive $\boldsymbol{E}^{\boldsymbol{n+1}}$. 4) Obtained matrix products $\boldsymbol{E}^{\boldsymbol{n}}$ and $\boldsymbol{E}^{\boldsymbol{n+1}}$, as certain types "sign matrices". 5) If necessary, to specify the pair signs ±$\boldsymbol{d.n.q.}_{ij}$ with the productive ±$\boldsymbol{d.n.q.e^{n}.}_{ij}$. RESULT: A specified sign matrix ***S*.***

<u>Comment:</u> **a)** The maximum number ***p*** can be greater than the number of differences among the semiotic pair signs and in some cases to *specify* these. **b)** Deeply identified pair sign, for example ±$\boldsymbol{d.n.q.e^{n}.}_{ij}$ we call to ***deep sign*** and corresponding matrix to ***deep sign matrix S*.*** **c)** Unfortunately it no works in case of strongly regular graphs where need use the *local identification* method P3.1.2.

<u>Example 3.2.</u> Result the Product IP. Product the adjacency matrices $\boldsymbol{E}\mathrm{x}\boldsymbol{E}\mathrm{x}\boldsymbol{E}=\boldsymbol{E}^3$, frequency vectors and adequacy of productive and semiotic pair signs of a regular graph**:**

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**=*i***	*12345*	***k=***
12	***18***	***16***	*13*	***19***	*13*	***16***	***18***	**1**	*12221*	***1***
18	*12*	***18***	***16***	*13*	***19***	*13*	***16***	**2**	*12221*	***1***
16	***18***	*12*	***18***	***16***	*13*	***19***	*13*	**3**	*12221*	***1***
13	***16***	***18***	*12*	***18***	***16***	*13*	***19***	**4**	*12221*	***1***
19	*13*	***16***	***18***	*12*	***18***	***16***	*13*	**5**	*12221*	***1***
13	***19***	*13*	***16***	***18***	*12*	***18***	***16***	**6**	*12221*	***1***
16	*13*	***19***	*13*	***16***	***18***	*12*	***18***	**7**	*12221*	***1***
18	***16***	*13*	***19***	*13*	***16***	***18***	*12*	**8**	*12221*	***1***

Productive e_{ij}^3	*12*	*13*	***16***	***18***	***19***
Semiotic dnq_{ij}	*0*	*-2.6.11*	***+2.5.8***	***+2.4.5***	***+2.4.6***

Comments: **a)** In present case prove the Product IP to sufficient for recognition the four classes of vertex pairs only in case $n=3$ **b)** Effectiveness of Product IP emerges in case of certain symmetric graphs, see Example 3.3.

Proposition 3.2. If graph ***G*** is ***vertex symmetric*** then for its canonical representation can be use a deep identification method.

Comments: **a)** In case of vertex symmetric graphs no arises the decompositions of sign matrix and for satisfy the conditions P2.2 need its using. **b)** It no mean that in each time it is necessary. For example, vertex symmetric graphs on Examples 1.2 (Petersen), 1.3 (DOD), 1.5 (Heawood), 2.2 (KOH) etc it does not require.

In following we look canonical presentations of the especially for isomorphism testing constructed two "very similar" *poly-symmetric graphs*. For specify the sign matrices used *Product Identification Principle*.

Example 3.4. Canonical presentations of *poly-symmetric graphs* PRA_A and PRA_B. Their common pair signs, deep pair signs and deep sign matrices:

Common basic pair signs of PRA_A and PRA_B:

A:-3.8.10; B:-3.6.7; C:-2.4.4; D:-2.3.2; ***E:+2.4.6****;* ***F:+3.8.16****.*

Specified by matrix product $E^{n=5}$ deep pair signs and deep sign matrix S_A* of graph PRA_A:

Marking the basic pair signs	*0*	*-A*	*-B*	*-C*		*-D*	***E***		***F***
Productive pair signs e^5	*180*	*125*	*110*	*165*	*160*	*80*	*231*	*233*	*210*
Marking the deep pair signs	*0*	*-A*	*-B*	*-C1*	*-C2*	*-D*	***E1***	***E2***	***F***

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	***i***	*ABCCD****EEF***	***Orb***
0	***E2***	***E1***	***E1***	***F***	*C2*	*C1*	*C1*	***F***	*C2*	*C1*	*C1*	*D*	*A*	*B*	*B*	*D*	*A*	*B*	*B*	**1**	*24422****212***	1
	0	***E1***	***E1***	*C2*	***F***	*C1*	*C1*	*C2*	***F***	*C1*	*C1*	*A*	*D*	*B*	*B*	*A*	*D*	*B*	*B*	**2**	*24422****212***	1
		0	***E2***	*C1*	*C1*	***F***	*C2*	*C1*	*C1*	***F***	*C2*	*B*	*B*	*D*	*A*	*B*	*B*	*D*	*A*	**3**	*24422****212***	1
			0	*C1*	*C1*	*C2*	***F***	*C1*	*C1*	*C2*	***F***	*B*	*B*	*A*	*D*	*B*	*B*	*A*	*D*	**4**	*24422****212***	1
				0	***E2***	***E1***	***E1***	*D*	*A*	*B*	*B*	***F***	*C2*	*C1*	*C1*	*A*	*D*	*B*	*B*	**5**	*24422****212***	1
					0	***E1***	***E1***	*A*	*D*	*B*	*B*	*C2*	***F***	*C1*	*C1*	*D*	*A*	*B*	*B*	**6**	*24222****212***	1
						0	***E2***	*B*	*B*	*D*	*A*	*C1*	*C1*	***F***	*C2*	*B*	*B*	*A*	*D*	**7**	*24222****212***	1
							0	*B*	*B*	*A*	*D*	*C1*	*C1*	*C2*	***F***	*B*	*B*	*D*	*A*	**8**	*24222****212***	1
								0	***E2***	***E1***	***E1***	*A*	*D*	*B*	*B*	***F***	*C2*	*C1*	*C1*	**9**	*24222****212***	1
									0	***E1***	***E1***	*D*	*A*	*B*	*B*	*C2*	***F***	*C1*	*C1*	**10**	*24222****212***	1
										0	***E2***	*B*	*B*	*A*	*D*	*C1*	*C1*	***F***	*C2*	**11**	*24222****212***	1
											0	*B*	*B*	*D*	*A*	*C1*	*C1*	*C2*	***F***	**12**	*24222****212***	1
												0	***E2***	***E1***	***E1***	*C2*	***F***	*C1*	*C1*	**13**	*24222****212***	1
													0	***E1***	***E1***	***F***	*C2*	*C1*	*C1*	**14**	*24222****212***	1
														0	***E2***	*C1*	*C1*	*C2*	***F***	**15**	*24222****212***	1
															0	*C1*	*C1*	***F***	*C2*	**16**	*24222****212***	1
																0	***E2***	***E1***	***E1***	**17**	*24222****212***	1
																	0	***E1***	***E1***	**18**	*24222****212***	1
																		0	***E2***	**19**	*24222****212***	1
																			0	**20**	*24222****212***	1

Specified by matrix product $E^{n=7}$ deep pair signs and deep sign matrix S_B* of graph PRA_B:

Basic pair signs	*0*	*-A*	*-B*		*-C*			*-D*	***E***		***F***
Productive signs e^7	*4410*	*3437*	*3276*	*3277*	*4081*	*4088*	*4011*	*3010*	*4831*	*4803*	*4445*
Deep pair signs	*0*	*-A*	*-B1*	*-B2*	*-C1*	*-C2*	*-C3*	*-D*	***E1***	***E2***	***F***

```
1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20|  i   ABBCCCDEEF  Orb
0 E1 E2 E1  F C1 C2 C3  F C3 C2 C1  D B2 B1  A  D  A B1 B2|  1   2222222212    1
   0 E1 E2 C3  F C1 C2 C1  F C3 C2  A  D B2 B1 B2  D  A B1|  2   2222222212    1
      0 E1 C2 C3  F C1 C2 C1  F C3 B1  A  D B2 B1 B2  D  A|  3   2222222212    1
         0 C1 C2 C3  F C3 C2 C1  F B2 B1  A  D  A B1 B2  D|  4   2222222212    1
            0 E1 E2 E1  D  A B1 B2  F C1 C2 C3  A  D B2 B1|  5   2222222212    1
               0 E1 E2 B2  D  A B1 C3  F C1 C2 B1  A  D B2|  6   2222222212    1
                  0 E1 B1 B2  D  A C2 C3  F C1 B2 B1  A  D|  7   2222222212    1
                     0  A B1 B2  D C1 C2 C3  F  D B2 B1  A|  8   2222222212    1
                        0 E1 E2 E1  A B1 B2  D  F C3 C2 C1|  9   2222222212    1
                           0 E1 E2  D  A B1 B2 C1  F C3 C2| 10   2222222212    1
                              0 E1 B2  D  A B1 C2 C1  F C3| 11   2222222212    1
                                 0 B1 B2  D  A C3 C2 C1  F| 12   2222222212    1
                                    0 E1 E2 E1 C3  F C1 C2| 13   2222222212    1
                                       0 E1 E2 C2 C3  F C1| 14   2222222212    1
                                          0 E1 C1 C2 C3  F| 15   2222222212    1
                                             0  F C1 C2 C3| 16   2222222212    1
                                                0 E1 E2 E1| 17   2222222212    1
                                                   0 E1 E2| 18   2222222212    1
                                                      0 E1| 19   2222222212    1
                                                         0| 20   2222222212    1
```

General and specified invariants and measures (values) of ***PRA_A*** and ***PRA_B***:

Symmetry	|V|	|R|	|E|	k	N^+	K	CL	MC	DM	SVV	SV	SEV^+	SE^+	TRA	BRA
Polysymm.	**20**	**190**	**50**	**1**	**3**	**1**	4	4	3	20^1	1.000	$10^1 20^2$	0.7303	0.250	0

G	N	P	SRV	HR	SR
PRA_A	8	8	$10^1 20^5 40^2$	0.8668	**0.6196**
PRA_B	10	10	$10^1 20^9$	0.9936	**0.5640**

Comments:

- **a)** Graphs ***PRA_A*** and ***PRA_B*** are ***non isomorphic.***
- **b)** Graph ***PRA_A*** differ at graph ***PRA_B*** by the number of pair(-)orbits, but coincide by pair(+)symmetric properties. In case of their complements it is in contrariwise.
- **c)** Both graphs are ***4-clique-, 4-girth-, 3-, 2-distance-*** and ***5-degree regular.*** *4*-clique regularity expressed by existence the five *4*-cliques, what are in sign matrix showy.
- **d)** Both graphs have three pair(+)orbits ***E1, E2*** and ***F*** with power 20.
- **e)** Graph ***PRA_A*** has five pair(-)orbits by *–A, -C2,* and *-D* with power 20 and two pair orbits by *-B* and *–C1* with power 40.
- **f)** The ***complement $PRAC_A$*** of ***PRA_A*** has pair signs *–A:-2.14.68, -B:-2.12.47,* ***C:+2.10.35, D:+2.10.36, E:+2.11.44, F:+2.12.48*** and is ***triangular*** and ***14-degree regular***.
- **g)** Graph ***PRA_B*** has seven pair(-)orbits with power 20.

On the graphs ***PRA_A*** and ***PRA_B*** give interest also their ***orbit structures*** that are more than *2-degee-regular.* Such are orbit structures of ***PRA_A*** by orbits *-B* and *–C1.*

Example 3.4. Sign matrix of ***orbit structure $PRA_{A(-B)}$*** of ***PRA_A*** by orbit *-B*:

A:-5.18.32; B:-4.8.12; C:-3.6.8; D:-2.6.8; E:-2.4.4; **F:+3.8.12.**

```
| 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20|  i  ABCDEF  Orb
  0 -D -A -A -E -E -C -C -E -E -C -C -B -B  F  F -B -B  F  F|  1  244144    1
     0 -A -A -E -E -C -C -E -E -C -C -B -B  F  F -B -B  F  F|  2  244144    1
        0 -D -C -C -E -E -C -C -E -E  F  F -B -B  F  F -B –B|  3  244144    1
           0 -C -C -E -E -C -C -E -E  F  F -B -B  F  F -B –B|  4  244144    1
              0 -D -A -A -B -B  F  F -E -E -C -C -B -B  F  F|  5  244144    1
                 0 -A -A -B -B  F  F -E -E -C -C -B -B  F  F|  6  244144    1
                    0 -D  F  F -B -B -C -C -E -E  F  F -B -B|  7  244144    1
                       0  F  F -B -B -C -C -E -E  F  F -B –B|  8  244144    1
```

```
0 -D -A -A -B -B  F  F -E -E -C –C|  9   244144   1
   0 -A -A -B -B  F  F -E -E -C –C| 10   244144   1
      0 -D  F  F -B -B -C -C -E -E| 11   244144   1
         0  F  F -B -B -C -C -E –E| 12   244144   1
            0 -D -A -A -E -E -C -C| 13   244144   1
               0 -A -A -E -E -C –C| 14   244144   1
                  0 -D -C -C -E -E| 15   244144   1
                     0 -C -C -E -E| 16   244144   1
                        0 -D -A -A| 17   244144   1
                           0 -A -A| 18   244144   1
                              0 -D| 19   244144   1
                                 0| 20   244144   1
```

Comments:

- **a)** Orbit structure $\boldsymbol{PRA_{A(-B)}}$ is (+)***symmetric, 5-partite*** and ***4-girth regular***.
- **b)** The parts of $\boldsymbol{PRA_{A(-B)}}$ correspond to *4*-cliques of $\boldsymbol{PRA_A}$: **I** – vertices **1,2,3,4; II** – vertices **5,6,7,8; III – 9,10,11,12; IV – 13,14,15,16**; **V – 17,18,19,20.** Part **I** is related with parts IV and V; **II** – with III and V; **III** – with II and IV; **IV** – with I and III; **V** – with I and II. In principle are the parts I and II addend to a part **A** and parts IV and V to a part **B,** but part III is not addend.
- **c)** Orbit structure $\boldsymbol{PRA_{A(-B)}}$ *coincide* with corresponding orbit structure of complement $\boldsymbol{PRAC_A}$ and is *isomorphic* with orbit structure by *–C1* of $\boldsymbol{PRA_A}$, i.e. isomorphic with $\boldsymbol{PRA_{A(-C1)}}$.
- **d)** Some characteristics of $\boldsymbol{PRA_{A(-B)}}$:

|V|	|R|	|E|	K	N	N^+	SVV	SV	SRV	HR	SR	SEV^+	SE^+	TRA	BRA
20	190	40	1	6	1	20^1	1.000	$10^1 20^2 40^4$	0.843	0.630	40^1	1.000	0	0

It is possible to construct bisymmetric and strongly regular graphs that need another deep identification method. At the same time is the probability of emerge such graphs go up to zero.

Example 3.5. Constructed by Netshepurenko et al [15] hardly distinguishable bisymmetric and strongly regular graphs $\boldsymbol{SIB_A}$ and $\boldsymbol{SIB_B}$ with 40 vertices have ***common pair signs***:

–A:–2.6.8 (on complement ***+B:+2.20.142***) and ***+B:+2.4.6*** (on complement *–A:–2.20.144*).

From pair signs conclude that $\boldsymbol{SIB_A}$ and $\boldsymbol{SIB_A}$ are *4-clique-*, *2-distance-* and *12-degree regular.* From coincidence the pair signs of $\boldsymbol{SIB_A}$ and $\boldsymbol{SIB_B}$ conclude the *coincidence of the symmetry properties.*

As in case the strongly regular graphs the Product IP (3.1.3) no works, then we must there to control the *pair signs of high degree pair graphs* to use the *high identification method* (P3.1.1). For $\boldsymbol{SIB_A}$ and $\boldsymbol{SIB_B}$ both obtain ***second degree pair signs***:

$–A^{m=2}=–3.18.48.$ and $+\boldsymbol{B}^{m=2}=\boldsymbol{+3.20.64}$,

that also no distinguish these graphs. A pair graph of third degree $\boldsymbol{g_{ij}}^{m=3}$ no arise, it is empty ∅. By $\boldsymbol{SIB_A}$ and $\boldsymbol{SIB_B}$ is touch with *very unusual* structures.

Now must be form by help the *local identification method* (P3.1.2) the *local sign matrices* $\boldsymbol{S_{ij}}^{m=2}$ for second degree pair graphs $\boldsymbol{g_{ij}}^{m=2}$ of $\boldsymbol{SIB_A}$ and $\boldsymbol{SIB_B}$. For this we open in both graphs a pair graph $\boldsymbol{g_{ij}}^{m=2}$, such that correspond to pair sign $+\boldsymbol{B}^{m=2}$.

Pair signs the local sign matrices $\boldsymbol{S_{ij}}^{m=2}$ of second degree pair graphs $\boldsymbol{g_{ij}}^{m=2}$ of $\boldsymbol{SIB_A}$ and $\boldsymbol{SIB_B}$ correspondingly:

Pair signs of second degree pair graph $\boldsymbol{g}^{m=2}\subset\boldsymbol{SIB_A}$ in local sign matrix $\boldsymbol{S_{ij}}^{m=2}{}_A$:

–A=–2.6.8; –B=–2.4.4; –C=–2.3.2; ***D=+2.4.6; E=+3.12.28; F=+3.20.46.***

Pair signs of second degree pair graph $\boldsymbol{g^{m=2}\subset SIB_B}$ in local sign matrix $\boldsymbol{S_{ij}{}^{m=2}{}_B}$:
–A=–2.6.8; –B=–2.4.4; ***C=+2.4.6; D=+3.12.24; E=+3.20.46.***

<u>Comments:</u> **a)** From differences of pair signs conclude *non-isomorphism* of second degree pair graphs. **b)** From non-isomorphism the pair graphs conclude non-isomorphism of graphs $\boldsymbol{SIB_A}$ and $\boldsymbol{SIB_B}$.

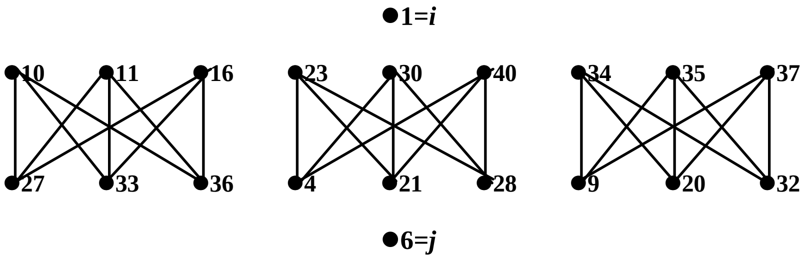
<u>Proposition 3.3.</u> From non-isomorphism the pair graphs $\boldsymbol{g_{ij}{}^A}$ and $\boldsymbol{g_{ij}{}^B}$ of corresponding symmetric graphs $\boldsymbol{G_A}$ and $\boldsymbol{G_B}$ conclude ***non-isomorphism*** of $\boldsymbol{G_A}$ and $\boldsymbol{G_B}$.

<u>Comment:</u> Consequently, the graphs $\boldsymbol{SIB_A}$ and $\boldsymbol{SIB_B}$ are non-isomorphic

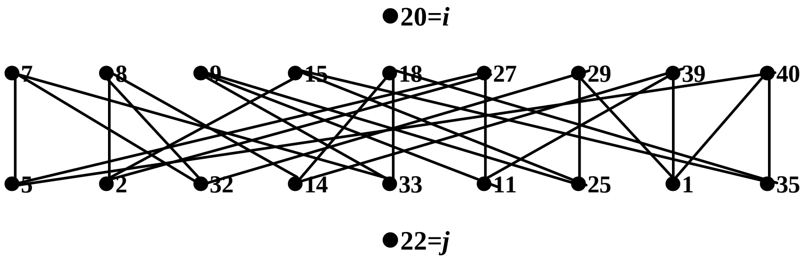
For showing the differences of $\boldsymbol{SIB_A}$ and $\boldsymbol{SIB_B}$ we demostrate second gegree pair graphs.

<u>Example 3.6.</u> The kernels of second degree pair graphs of very similar structures $\boldsymbol{SIB_A}$ and $\boldsymbol{SIB_A}$:

Kernel of $\boldsymbol{g_{1\text{-}6}{}^{m=2}\subset SIB_A}$:



Kernel of $\boldsymbol{g_{20\text{-}22}{}^{m=2}\subset SIB_B}$:



3.3. Graph isomorphism problem in general

Isomorphism (Greek word *isos* – same; *morphe* – form) constitutes *philosophical category* that mean *one-to-one correspondence* between *structures* of objects [6]. Such a one-to-one correspondence can exist only between abstract, idealized objects.

In *mathematics* we define isomorphism as a one-to-one mapping of one system into another same type system, which preserves the structure, i.e. relations, ordering, topology etc. Isomorphism is *an invertible morphism*, which has *an opposite morphism*, such that their product is *the unity morphism*. A topological isomorphism is called a *homeomorphism*. The ***isomorphism problem*** is to design an algorithm that recognizes the isomorphism of two objects. To ***canonical presentation*** of an object we call a formation that in a concentrated form show or describe it.

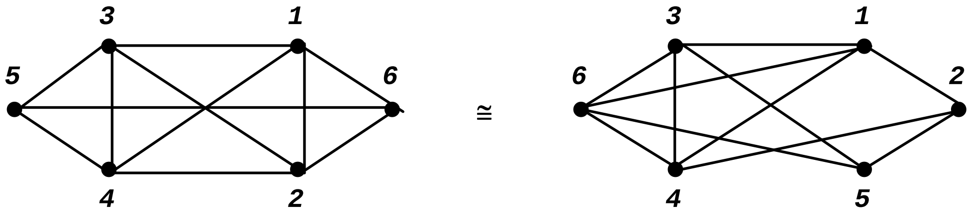
The graph isomorphism problem first came into prominence in 1857, when Arthur Cayley [16] reported his research on organic isomers. Subsequently, isomorphism became a central problem in graph theory. According to the “2000 Mathematics Subject Classification” (MSC2000), isomorphism recognition together with the reconstruction problem, is a combinatorial phenomenon 05C60, even when graph theory itself is not yet classified as a separate subject!

Two graphs are called isomorphic, if they differ only in the labeling of their vertices. An isomorphic mapping from graph G_A to graph G_B is an isomorphic substitution $\varphi: V_A \to V_B$

Proposition 3.4. ***The classical isomorphism recognition*** is an answer to the question, is graph G_A isomorphic to graph G_B? If so, one must also provide the isomorphic substitution.

A naïve algorithm for isomorphism recognition obviously exists – try all possible substitutions (permutations) of the rows and columns of the adjacency matrix of G_B until it coincides with adjacency matrix of graph G_A. However, this is an impossible task to perform for all practical purposes, since the number of permutations that one may need to check can go up to $n!$ (n-factorial). For example, checking 16! permutations could take up to 40 years of time on the fastest computers presently available.

Example 3.7. Are graphs G_A and G_B isomorphic?



1	0 1 1 1 0 1	4	0 1 1 1 0 1	1→4
2	1 0 1 1 0 1	1	1 0 1 1 0 1	2→1
3	1 1 0 1 1 0	3	1 1 0 1 1 0	3→3
4	1 1 1 0 1 0	6	1 1 1 0 1 0	4→6
5	0 0 1 1 0 1	5	0 0 1 1 0 1	5→5
6	1 1 0 0 1 0	2	1 1 0 0 1 0	6→2

Graphs G_A and G_B turn out to isomorphic, but displacing mode the rows and columns of adjacency matrix ask there 6!=720 steps.

We must then begin to seek other, more useful ways for isomorphism testing. Ideally, we should *try to find a polynomial-time algorithm* for the graph isomorphism problem. That is, we should *try to show that the graph isomorphism problem in* ***P***. During the 1970’s, this was a very popular research activity. For example, S. Toida [17] presents for this purpose the concept of a “distance matrix”. Non-isomorphism can be recognized by distance matrices “almost always”, and isomorphism testing is possible in many cases, but not always.

Algorithms of this period were heavily criticized by R. C. Read, D. G. Corneil [18] and G. Gati [19], who called this “hobby” the “isomorphism disease”. The isomorphism problem became taboo during this period. This problem is avoided in some graph textbooks to this day. For example, B. Bollobas’ “Modern Graph Theory” [20] dedicates only two words for the isomorphism problem. Nevertheless, a small visual example of graph isomorphism is presented in almost all textbooks – and nothing more is said.

There are numerous monographs dedicated specially to the isomorphism problem. The aspect of group theory was treated by C. Hoffmann [21], who asserted that the "structure" of groups was quite similar to the general isomorphism problem. Unfortunately, this similarity turned out to be elusive. The isomorphism problem is treated in great depth by Netchepurenko et al [15] where they also present corresponding algorithms and computer programs that work "almost always". In the monograph by G. Köbler, H. Schönig and J. Toran [22] this problem is treated on the basis of structural complexity.

Without good algorithms, the treatment of the isomorphism problem is senseless. Some partial progress was made by L. Babai [23], who found that in certain cases a Monte-Carlo algorithm is suitable. G. Tinhofer, M. Lödeke, S. Baumann, L. Babel [24] feel convinced that isomorphism testing is solvable by Weisfeiler-Leman algorithm. C. V. Raj and M. S. Shivakumar [25], give some attributes for solving the problem under certain conditions. At the same time, some monographs of algorithmic graph theory, such as N. Chistofiedes [26], have nothing to say about isomorphism. S. Pemmaraju and S. Skiena [27] are limited to studying time complexity of the isomorphism problem.

The isomorphism problem has mostly been studied just on the complexity aspect [28]. It is not known whether this problem is **NP**-complete. Whereas nobody found any polynomial-time solutions in **P** before us, it has been presented as an intermediary variant, marked **SPP** [29].

The methods of isomorphism recognition can be divides to: a) sorting or non-sorting methods; and, b) methods with using local or global invariants.

On the structural aspect we can be say that graphs G_A and G_B are isomorphic *if and only if these have one and the same structure.*

3.4. Isomorphism recognition by sign matrices

An ***invariant*** is an *attribute* of an object, such as a size, form of an expression etc., which *stays unchangeable* in case of certain transformations, i.e. it is an invariant in relation to these transformations. For example, *local invariants* can be degrees of the vertices, distances between vertices, pair signs etc. Similarly, *global invariants* can be a degree frequency vector, various codes, polynomials, spectrums etc. of a graph. If the observed transformation does not change some attribute of an object, then it is a *complete invariant* [30]. For example, a complete invariant of isomorphic graphs is their common structure that does not change in case of relabelling (remarking) and/or transposition of their vertices.

The concept of invariants has been used in mathematics since the middle of the 19th century. Invariant theory had great importance in geometry. Invariant theory is treated classically as an algebraic theory [31]. Later, this specific concept was promoted to a *philosophical category.*

According to F. Harary [30], the isomorphism problem is solvable by complete system of global invariants (polynomials, spectra) of graphs. S. Locke [13] found that 3-cube-codes (super-long binary codes) are sometimes useful for isomorphism testing. From A. Zykov's point of view [32], the isomorphism problem is solvable on the basis of a system of local invariants that characterize the compactness, cycles (girths), paths etc., of a graph.

Isomorphism problem is solvable also by comparison the sign matrices S_A and S_B.

Structural Equivalence Principle. OPERAND: *Decomposed sign matrices* S_A *and* S_B. ALGORITHM: To control the *structural equivalence* $S_A \approx S_B$, i.e: a) *coincidence* the pair signs $\{\pm d.n.q._{ij}\}_A$ and $\{\pm d.n.q._{ij}\}_A$; b) *coincidence* the frequency- $\{u_i\}_A = \{u_i\}_B$ and class vectors $\{s_i\}_A = \{s_i\}_B$ in the framework of each vertex class $W_k \subset W$, $k \in [1, K]$. RESULT: *Equivalence* $GS_A \equiv GS_B$ *or non-equivalence of the structures,* i.e. *isomorphism* $G_A \cong G_B$ *or non-isomorphism the graphs.*

Example 2.2. Decomposed sign matrices S_A and S_B of graphs G_A and G_B that are showed on Example 2.1.:

```
A:-2.5.7; B:+2.3.3; C:+2.4.6; D:+2.5.8; E:+3.6.11.

| 1   1   1   1| 2   2|     uᵢ     sᵢ k      | 1   1   1   1| 2   2|     uᵢ     sᵢ k
| 1   2   3   4| 5   6| i  ABCDE   12 .      | 1   3   4   6| 2   5| i  ABCDE   12 .
| 0   D   C   C|-A   B| 1  11210   31 1      | 0   C   D   C| B  -A| 1  11210   31 1
      0   C   C|-A   B| 2  11210   31 1            0   C   D|-A   B| 3  11210   31 1
          0   D| B  -A| 3  11210   31 1   ≈            0   C| B  -A| 4  11210   31 1
              0| B  -A| 4  11210   31 1                    0|-A   B| 6  11210   31 1
               | 0   E| 5  22001   21 2                     | 0   E| 2  22001   21 2
                     0| 6  22001   21 2                           0| 5  22001   21 2
```

Comments: **a)** Sign matrices of graphs G_A and G_B are *structurally equivalent*, $S_A \approx S_B$. Consequently, graphs are isomorphic $G_A \cong G_B$. **b)** Structural equivalence of sign matrices, $S_A \approx S_B$ is *one-to-one correspondence* the partial matrices $(W_{kk'})_A \leftrightarrow (W_{kk'})_B$, pair orbits $(\Omega R_n)_A \leftrightarrow (\Omega R_n)_B$ and their corresponding pair signs $dnq_A \leftrightarrow dnq_B$.

Proposition 2.2. ***Equivalence of structures*** $GS_A \equiv GS_B$ constitute an isomorphism $G_A \cong G_B$ with exactness up to components, parts, vertex orbits $(\Omega V_k)_A \cong (\Omega V_k)_B$, pair orbits $(\Omega R_n)_A \cong (\Omega R_n)_B$, orbit graphs $(G_n)_A \cong (G_n)_B$, pair graphs $(g_{ij})_A \cong (g_{ij})_B$, etc.

Comment: Structural equivalence no needs isomorphism recognition on the aspect of vertex substitutions.

3.5. Canonical outputs of isomorphism algorithms

Only few isomorphism recognition algorithms give a canonical output of processing results. Usually be limited laconically with phrase “isomorphic” or “not isomorphic”. We show the ***canonical output*** of two algorithms the isomorphism recognition. Recognition the orbits no belong to isomorphism problem.

In our historical journey so far, the question remains: *is the graph isomorphism problem in* ***P****?* Clearly, some essential ideas are still missing.

The correct ***polynomial algorithm*** of Ashay Dharvadker et al [8] be grounded on the formation of incomplete sign matrices S_A ja S_B. These are decomposed up to frequency classes. In the framework of these classes take place rearrange the rows ***i*** and columns ***j*** to ***isomorphism recognition with exactness up to substitutions.*** The time complexity and recognition self are proved in detail. The results are excellently disained. On the classical view point it is the *general isomorphism recognition algorithm* in structure semiotics.

Example3.9. Canonical outputs of Ashay Dharwadker´s et al algorithm for graphs G_A and G_B:

Sign matrix of G_A (I output):

Matrix *A*	**4**	**5**	**1**	**2**	**3**	**7**	**8**	**6**
4	-0.1.0	-2.8.21	+2.5.7	+2.5.7	+2.5.7	+2.5.7	+2.5.7	+2.5.7
5	-2.8.21	-0.1.0	+2.5.7	+2.5.7	+2.5.7	+2.5.7	+2.5.7	+2.5.7
1	+2.5.7	+2.5.7	-0.1.0	-2.7.16	-2.7.16	+2.4.5	+2.4.5	+2.4.5
2	+2.5.7	+2.5.7	-2.7.16	-0.1.0	-2.7.16	+2.4.5	+2.4.5	+2.4.5
3	+2.5.7	+2.5.7	-2.7.16	-2.7.16	-0.1.0	+2.4.5	+2.4.5	+2.4.5
7	+2.5.7	+2.5.7	+2.4.5	+2.4.5	+2.4.5	-0.1.0	-2.7.16	-2.7.16
8	+2.5.7	+2.5.7	+2.4.5	+2.4.5	+2.4.5	-2.7.16	-0.1.0	-2.7.16
6	+2.5.7	+2.5.7	+2.4.5	+2.4.5	+2.4.5	-2.7.16	-2.7.16	-0.1.0

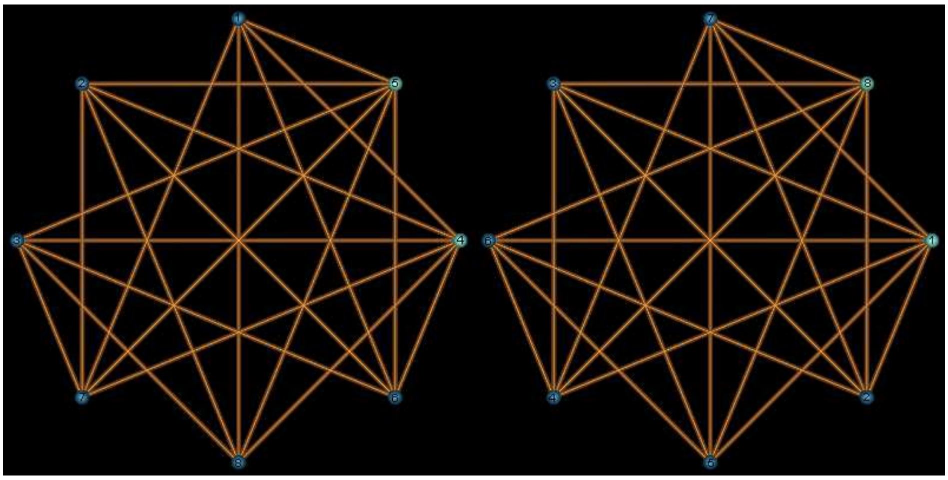
Sign matrix of G_B (II output):

Matrix *B*	**1**	**8**	**7**	**3**	**6**	**4**	**5**	**2**
1	-0.1.0	-2.8.21	+2.5.7	+2.5.7	+2.5.7	+2.5.7	+2.5.7	+2.5.7
8	-2.8.21	-0.1.0	+2.5.7	+2.5.7	+2.5.7	+2.5.7	+2.5.7	+2.5.7
7	+2.5.7	+2.5.7	-0.1.0	-2.7.16	-2.7.16	+2.4.5	+2.4.5	+2.4.5
3	+2.5.7	+2.5.7	-2.7.16	-0.1.0	-2.7.16	+2.4.5	+2.4.5	+2.4.5
6	+2.5.7	+2.5.7	-2.7.16	-2.7.16	-0.1.0	+2.4.5	+2.4.5	+2.4.5
4	+2.5.7	+2.5.7	+2.4.5	+2.4.5	+2.4.5	-0.1.0	-2.7.16	-2.7.16
5	+2.5.7	+2.5.7	+2.4.5	+2.4.5	+2.4.5	-2.7.16	-0.1.0	-2.7.16
2	+2.5.7	+2.5.7	+2.4.5	+2.4.5	+2.4.5	-2.7.16	-2.7.16	-0.1.0

Substitution table for graphs G_A and G_B (III output):

Graph G_A	**Graph G_B**
4	**1**
5	**8**
1	**7**
2	**3**
3	**6**
7	**4**
8	**5**
6	**2**

Graphs G_A and G_B are isomorphic (IV output):



By help of Ashay Dharwadker´s et al isomorphism algorithm are recognized also canonically hardly recognizable graphs. For example, the non-isomorphism of graphs SIB_A and SIB_B (see Example 3.5) are recognized with exactness up to substitutions and their non-isomorphism is visually showed.

*

Isomorphism algorithm of Blazej Podsiadlo [33] is also polynomial. To canonical output of a graph is its *biggest value* that no contain data about the graph, but enable to differentiate these, better as for example 3-cube-codes. It do no realized up to substitutions. To the canonical output belong *the biggest value* <**the biggest value**>, *the number of paths* <**paths**>, *the number of automorphisms* <**automorphisms**>, *the real time* <**treal**>.

<u>Examples 3.10.</u> Some results of isomorphism algorithm, by Blazej Podsiadlo**:**

<u>Comment:</u> The ***biggest values*** by Blazej Podsiadlo for isomorphism testing and their using for ***measure the isomorphism***, rather ***the similarity*** of graphs**.**

<example> <paths> <automorphisms> <treal> <the biggest value>

<1A> <120> <120> <0m0.014s>
555274755282898162809472913496
<**1B>** <**120> <120> <0m0.021s>**
10805731416208407477923644473991185184903230
Result: **NOT Isomorphic**
It ensues on the complete differences of the values and their “lengths”**.** It is a performance with the biggest values of **Petersen’s graph** (10 vertices) and **icosahedron** (12 vertices).

<2A> <5408> <2> <0m0.191s>
69600325658926472221667080008746713668146791061252264141364537596668601347071669
20510329885036595840342918872926941148691479059618393408947807324858975660236112
0526675494159293310332512602
<2B> <5972> <3> <0m0.217s>

696003256589264700591756491225913747187438885125701195348138731649257570771478130
05228568388843713716865188561842593294496128281847460135789592203908404395481627
6553574432763616146379938138
Result: **NOT Isomorphic.**
It ensues on the very small coincidence in the beginning of sums **16/188** or **8,51%** similar**.** It is a performance with two strongly regular **Weisfeiler's graphs** [14, p 166] that have common pair signs.

<3A> <7464960> <51840> <2m57.011s>
2222578069750096457096546578094317441414831428937332944527939242242896479846933821042719
405754577875110179721773304399585144313227788979663142023016421504216767
86803877725700271887954588059323245147959799837063854397147649717283678693799805
73103179047818966387304387830423283405591144366704227236711352393085980534919893
92714049494008725024901283676435179374075384660700164314246730083748936847275009
78533365931549076597669812253320077432965869201259728770624981891511746224042212
00
<3B> <1244160> <51840> <4m43.816s>
2222578069750096457096546578094317441414831428937332944527939242242896479846933821042719
3208813316900503614212365104449298967747059843579229784513714819169978033
74219353382974233990523835794385357692050347712780681876020610457624122754550464
11313074785929077002108862539314258558246216122361441977447436931023241071626921
18737276609400559158153127807996209487334667099023709188011542186468673965530522
21917772479448327901335088290550331691343285639201843291261639337656323940805736
00
Result: **NOT Isomorphic.**
It ensues also on the small coincidence in the beginning of sums **88/482** or 1**8,25%** similar**.** It is a performance with two strongly regular **Klin's graphs *SIB*** [15] that have common pair signs and their difference is elusory.

<4A> <7200> <47> <0m0.217s>
68529550488314234072213829872632606950390825315001971666039649567375373877065894
63002605803964759814670962424580757192086313188244219551300422275403305788480722
7857230452232597583140280320
<4B> <7200> <23> <0m0.199s>
68529550488314234072213829872632606950390825315001971666039649567375373877065894
63002605803964759814670962424548552518226958640467373781724797655462921343320362
4911285433408634651150440448
Result: **NOT Isomorphic**
It ensues on the partial coincidence in the beginning of sums **110/188** or 5**8,51%** similar**.** It is a performance with two bipartite **Mathon's graphs** [15, Table I] that have common first degree pair signs and are rather similar.

<5A> <720> <79> <0m0.074s>
12507790066141064298704970931787602012827179545680434511465325842835586260004765
35653120832518219913637860703695905227100
<5B> <720> <34> <0m0.074s>
12507790066141064298704970931787602012827179545680434511465325842835586260004765
35621219280757394118382351094747106378100
Result: **NOT Isomorphic**
It ensues on the rather great coincidence in the beginning of sums **83/121** or **68,60%** similar**.** It is a performance with two vertex symmetric graphs ***PRA***$_A$ and ***PRA***$_B$ that have common first degree pair signs and are very similar.

<6A> <138> <135> <0m0.080s>
4953860357280524777774009458554957194272212680148363306943060955749799792961656953336056
<6B> <137> <135> <0m0.086s>
4953860357280524777774009458554957194272212680148363306943060955749799792961656953336056
Result: **Isomorphic.**
It ensues on the complete coincidence of sums or **100%** similar**.** It is a performance with two **Ramsey's graphs** that have the same structure but that are labeled differently.

As we see, the "length" of value depends on the vertex number and coincidence on relation the "lengths" of intersection and full value. It can be treat as an "isomorphism measure". Naturally, their essence needs to research.

4. THE PROBLEMS OF ADJACENT STRUCTURES AND RECONSTRUCTIONS

To adjacent structures we call the greatest sub-graphs and smallest super-graphs of a graph that are related with reconstruction problem. The reconstruction problem is well-know as the *Ulam's Conjecture* [7].

4.1. Relationships between isomorphic graphs and their ($G\backslash v_i$)-sub-graphs

Here is suitable to begin with a theorem, proved by A. Titov in 1975 [35].

Titov theorem If all the ($G\backslash v_i$)-sub-graphs of graph G are isomorphic, then automorphism group *Aut***G** is transitive on the set of vertices V.

Comment: It mean that graph G *is vertex symmetric ("transitive"),* i.e. *there exists only one vertex orbit* $\Omega V_{k=1=K}$ which correspond just to one isomorphism class $\Gamma_{k=1=K}$ of ($G\backslash v_i$)-sub-graphs.

There we return anew to problems of the relationships between *equal positions* and *remaining graphs* (Prop. 2.1) and relationships between *automorphisms, local isomorphisms, orbits* and *semiotic invariants* (Prop. 2.2).

Indeed, vertex orbit $\Omega V_{k=1=K}=\Omega(v_{i=1},\dots,v_{i=|V|})_{k=1=K}$ is *a transitivity domain of automorphisms in* ***AutG*** that be expressed by an isomorphism class $\Gamma_{k=1=K}$ of ($G\backslash v_i$)-sub-graphs and is represented in the form of an *equivalence class* of vertices in partial matrix S_k of decomposed sign matrix S.

Example 4.1. Sign matrix S_A of the graph G_A (see also Example 2.9):

A:-2.8.21; B:-2.7.16;
C:+2.4.5; D:+2.5.7