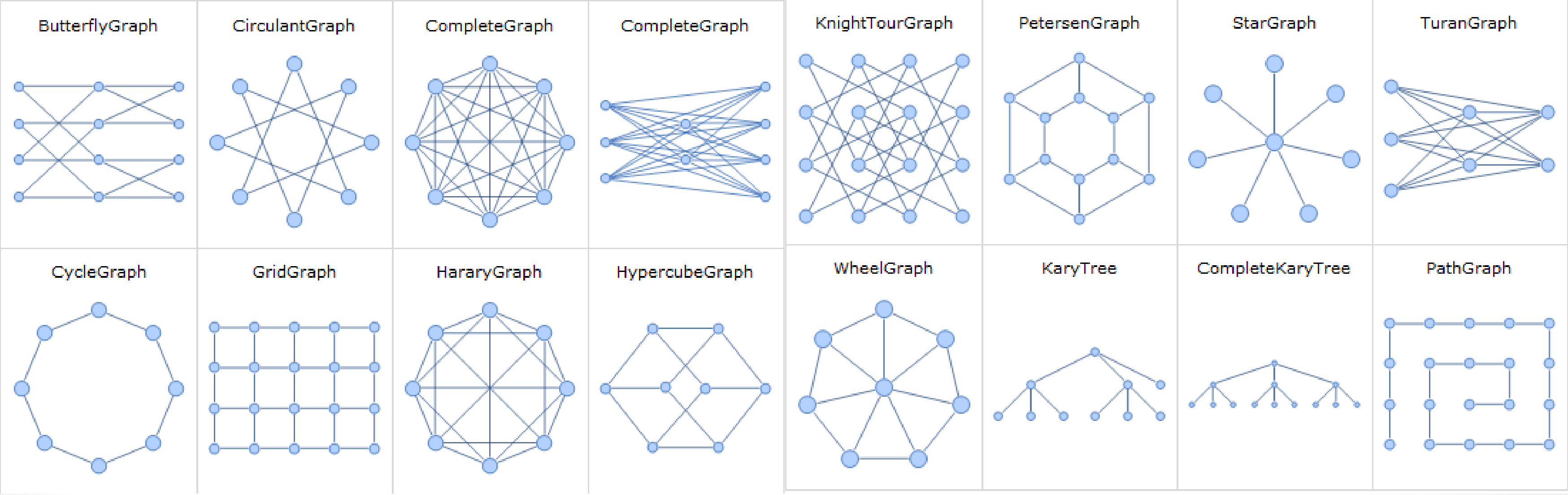
Irregular Labeling of Graph Models using Algorithmic Approach

Research Directions

- Studying different types of graph labeling
- Studying different families of graphs



Research Directions

- Studying different types of graph labeling
- Studying different families of graphs
- Devising new algorithms for labeling as exact or improved bounds
- Applying algorithms on computers for accurate results to compare them with mathematical results

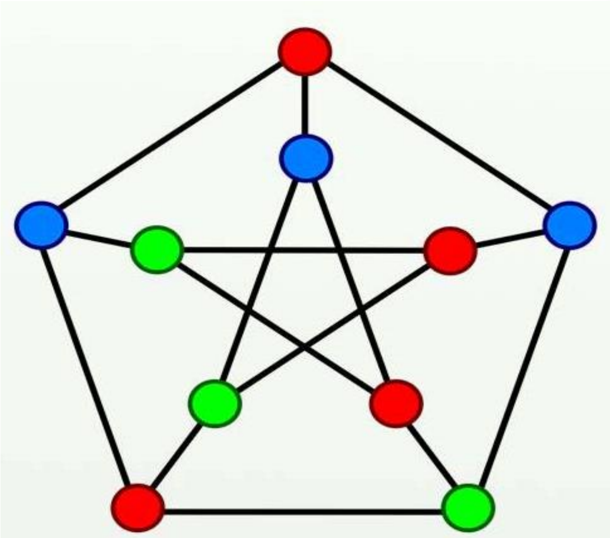
Graph Labeling

An assignment of integers to the vertex set or edge set, or both.

Three common characteristics:

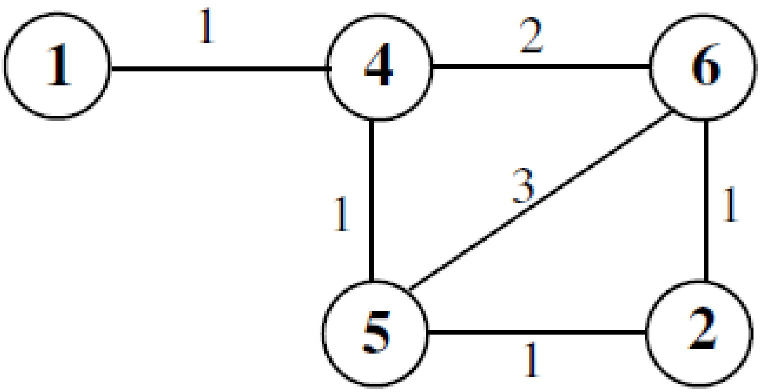
- A set of integers as labels
- A rule that assigns the labels
- Some condition(s) that these labels must satisfy.

Graph coloring is a special case of graph labeling, it is an assignment of labels traditionally called "colors" to elements of a graph. If no two adjacent vertices are of the same color this is called a vertex coloring. Similarly, an edge coloring assigns a color to each edge so that no two adjacent edges are of the same color.



Irregularity Strength : s(G)

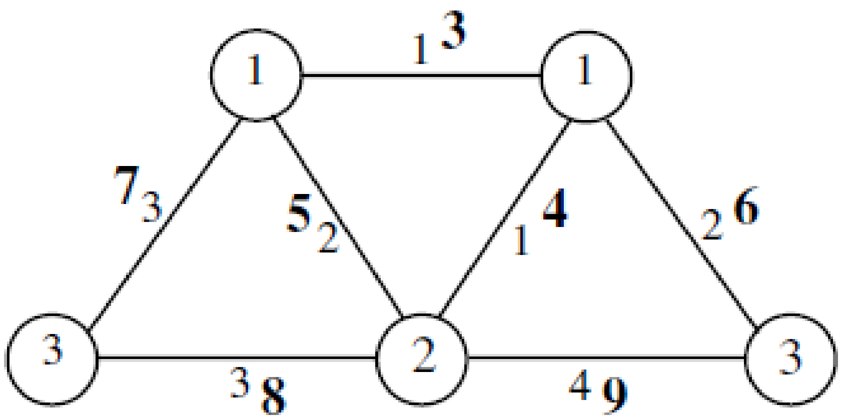
In 1988 **Chartrand** introduced edge *k*-labeling of a graph such that $w_\varphi(x) \neq w_\varphi(y)$ *for* all vertices $x, y \in V(G)$. Such labeling was called irregularity strength ***s(G)*.**



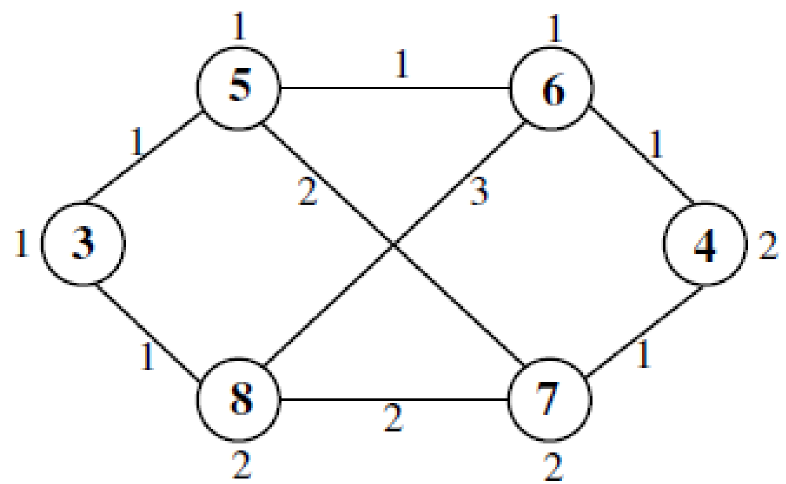
G. Chartrand, M.S. Jacobson, J. Lehel, O.R. Oellermann, S. Ruiz and F. Saba,
Irregular networks,
Congr. Numer. 64 (1988), 187–192.

Edge Irregular Total k-Labeling and Vertex Irregular Total k-Labeling

- In 2007, **Baca** introduced two types of total k-labelings:
  - ➢ Edge irregular total k-labeling tes(G)
  - ➢ Vertex irregular total k-labeling tvs(G)



Example for tes(G)



Example for tvs(G)

Ba˘ca, M., Jendro˘l , S., Miller, M. and Ryan, J.,
On irregular total labelings,
Discrete Math. 307 (2007), 1378–1388.

Edge Irregularity Strength: *es*(G) or Vertex *k*-labeling

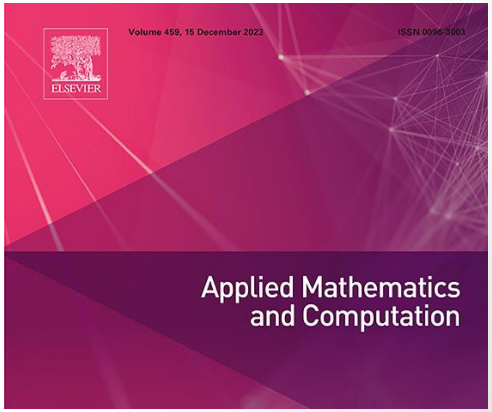
In 2014 **Ahmad** introduced vertex *k*-labeling

$$\phi : V(G) \rightarrow \{1, 2, \ldots\ldots, k\}$$

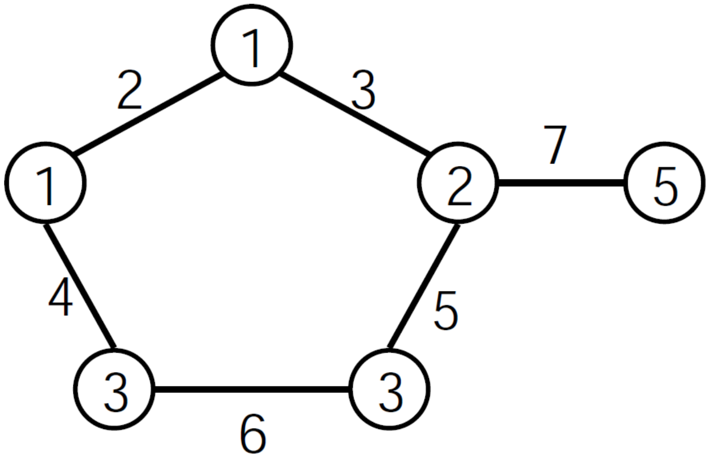
- For every two different edges *e* and *f* $\quad w_\phi(e) \neq w_\phi(f)$
- Edge weights: $\quad w_\phi(xy) = \phi(x) + \phi(y)$
- Minimum ***k*** is called the edge irregularity strength ***es(G)***.

$$es(G) \geq \max\left\{ \left\lceil \frac{|E(G)|+1}{2} \right\rceil, \Delta(G) \right\}$$

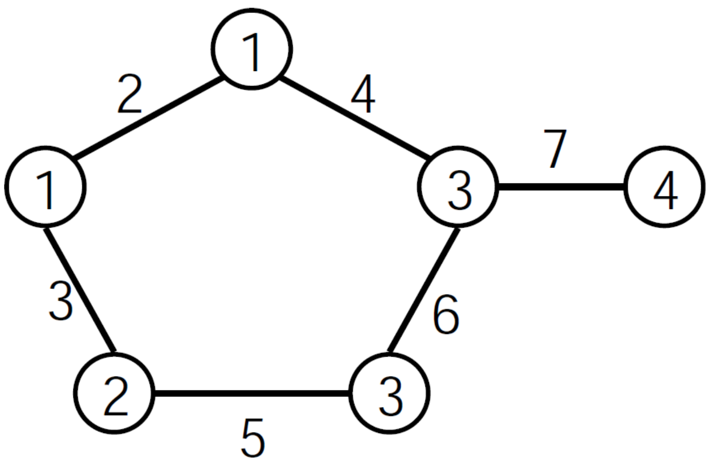
Ahmad, O. Al-Mushayt, M. Baca,
On edge irregularity strength of graphs,
Appl. Math. Comput. **243**(2014), 607–610



Vertex *k*-labeling : Edge Irregularity Strength *es(G)*



V(G) = 6
E(G) = 6
Labels = {1,1,2,3,3,**5**}
Max Edge Weight = 7



V(G) = 6
E(G) = 6
Labels = {1,1,2,3,3,**4**}
Max Edge Weight = 7

$$es(G) \geq \max\left\{ \left\lceil \frac{|E(G)|+1}{2}, \Delta(G) \right\rceil \right\}$$

Algorithmic Solutions

- Algorithms can solve many problems, where other mathematical solutions are very complex / impossible.

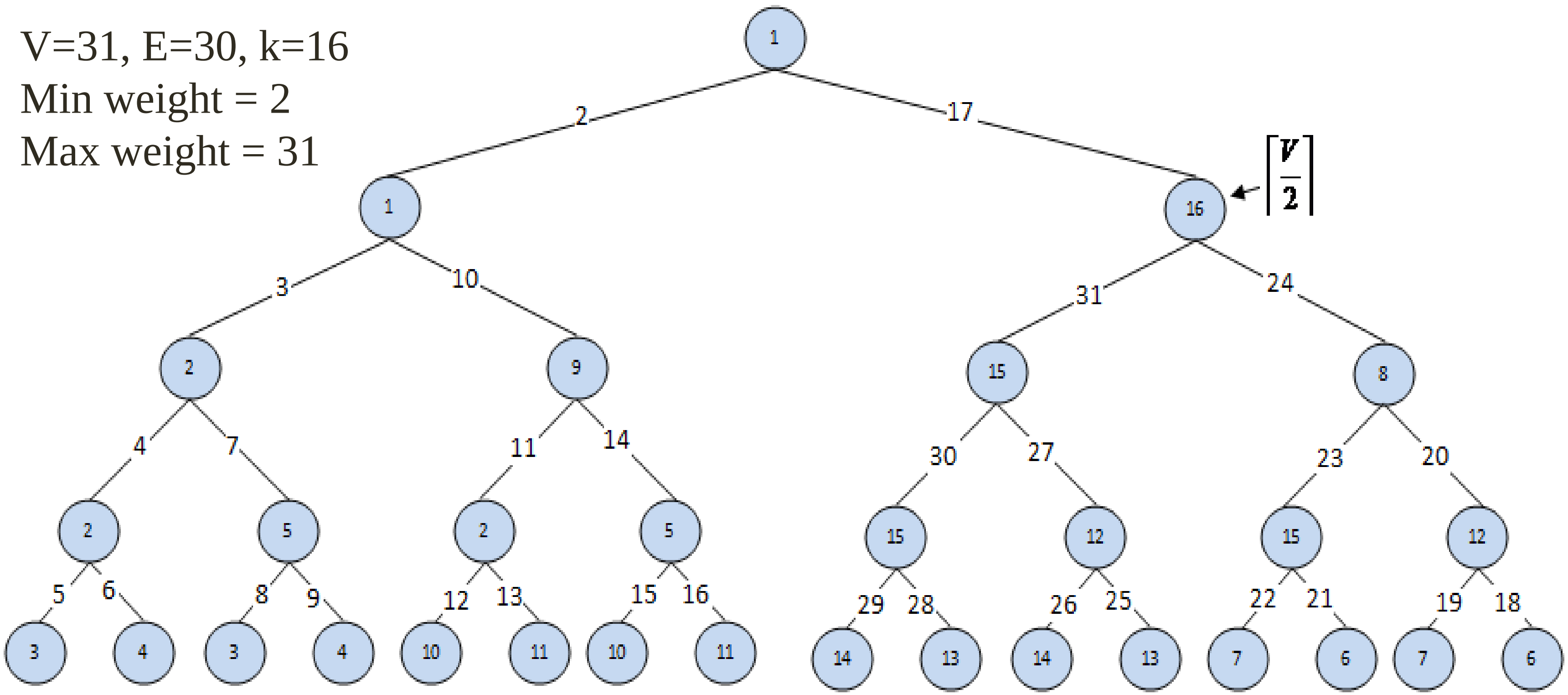
- Computations can tackle the exhaustive issues of numerical calculations by providing comprehensive results.

- Algorithmic Design Strategies (Helpful in Assig-3)
  - Brute Force
  - Divide & Conquer
  - Backtracking
  - Dynamic Programming

Results of Computer Based Experiment

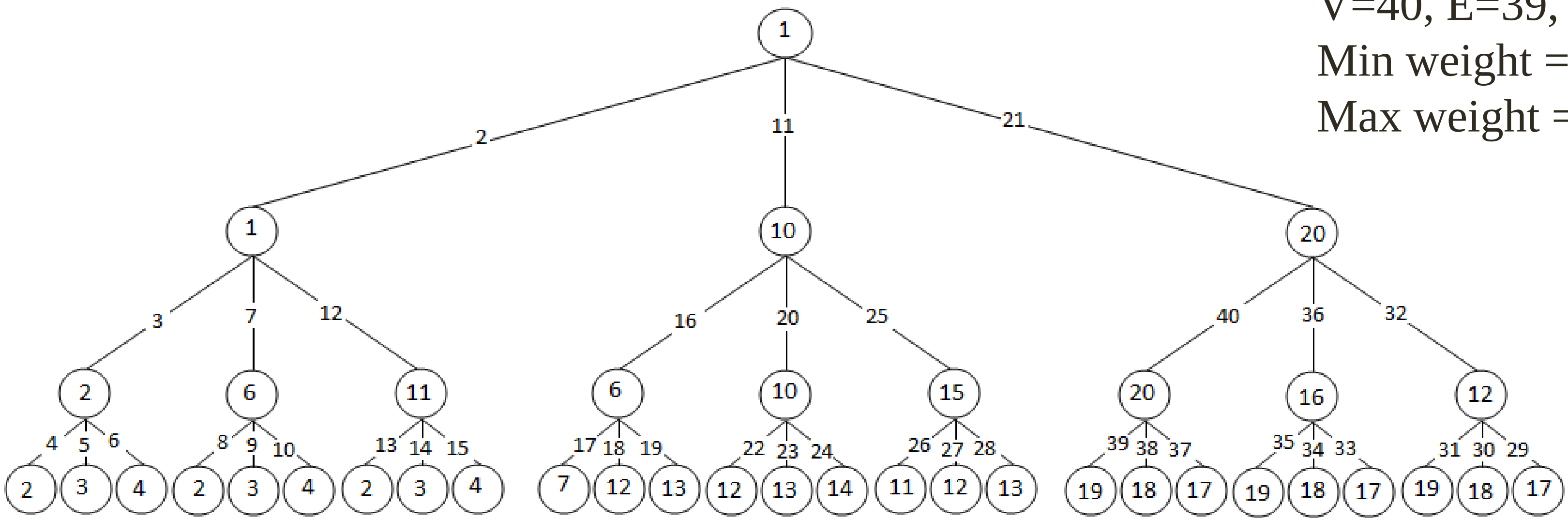
- Value of ***k*** is exactly $= \left\lceil \frac{V}{2} \right\rceil$
- Value of ***k*** always found at ***mth*** child of root vertex.
- Left tree follows ascending order whereas right tree follows descending order.

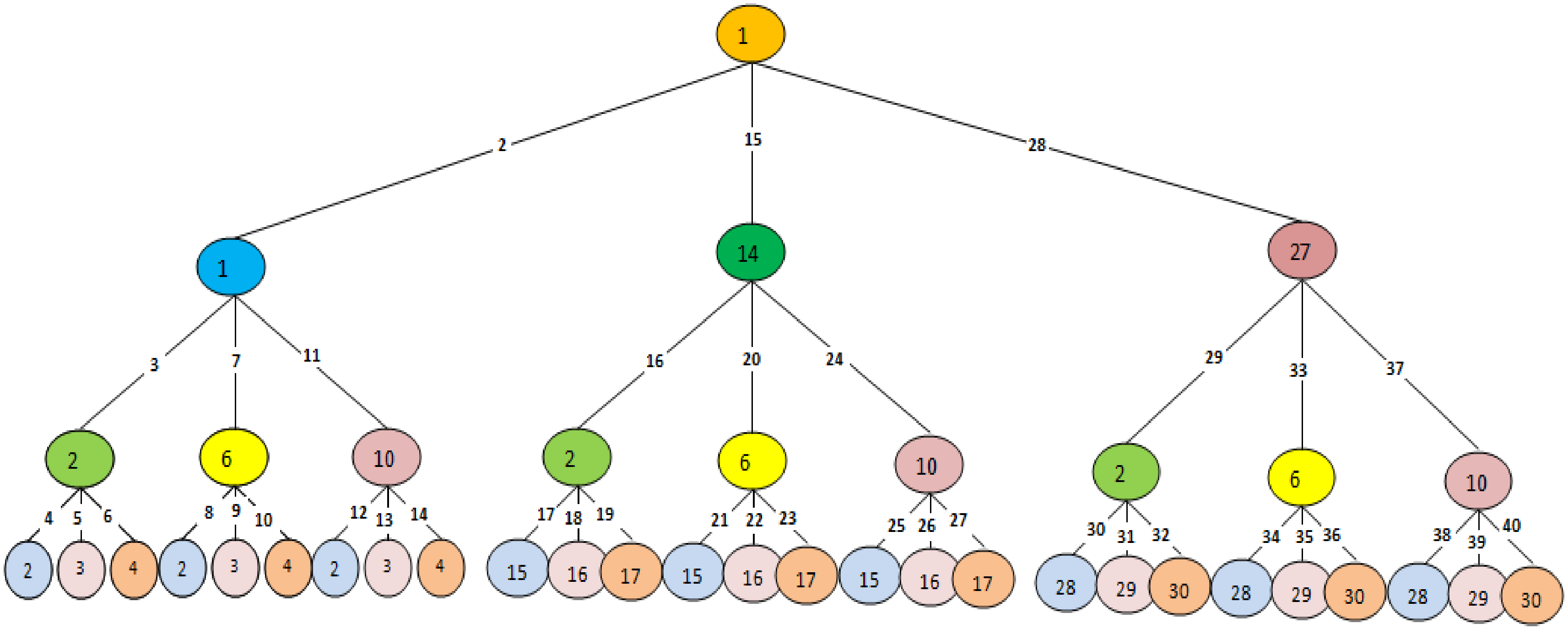
V=31, E=30, k=16
Min weight = 2
Max weight = 31



Complete Ternary Tree :($T_{3,3}$)

V=40, E=39, k=20
Min weight = 2,
Max weight = 40

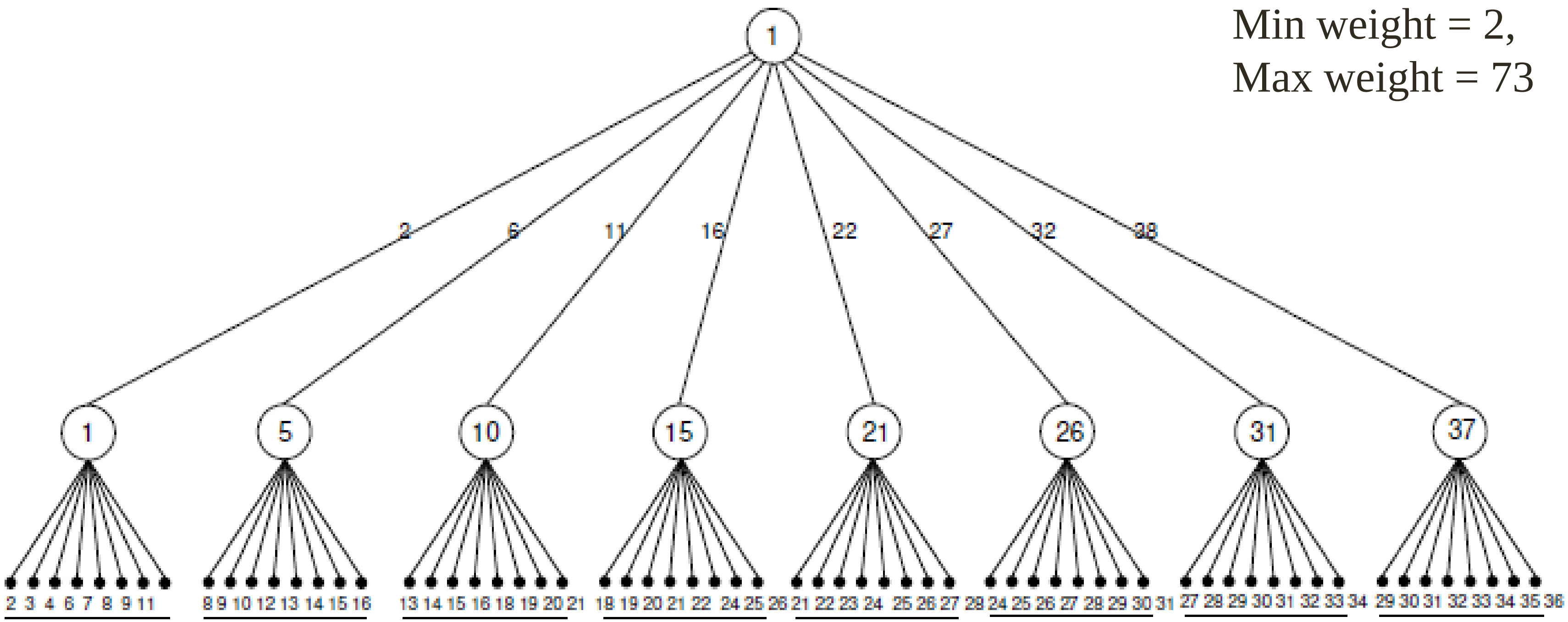




11 March 2024

M-ary Complete Trees

V=73, E=78, k=37
Min weight = 2,
Max weight = 73



$$d = \left\lfloor \frac{\left\lceil \frac{V}{2} \right\rceil}{m-1} \right\rfloor$$

Level 1	Childs of root	Left most	2	3	....	m^{th}
	Labels	1	$val_1 = 0 + d$	$val_2 = val_1 + d$	….	k
	Edges Weights	2	$val_1 + 1$	$val_2 + 1$	….	k+1

A. Ahmad, M. Baca, M.A. Asim, R. Hasni, "Computing Edge Irregularity Strength of Complete m-ary Trees using Algorithmic Approach", UPB Scientific Bulletin, Applied Mathematics and Physics 80(3), 2018.

Complete Graphs: es(Kn)

- Complete graph is a famous type of regular graphs denoted as K_n.
- In complete graph every two distinct vertices are adjacent.
- Complete graph K_n is a regular graph of degree $r = n-1$ and size is $K_n = \frac{(n(n-1))}{2}$

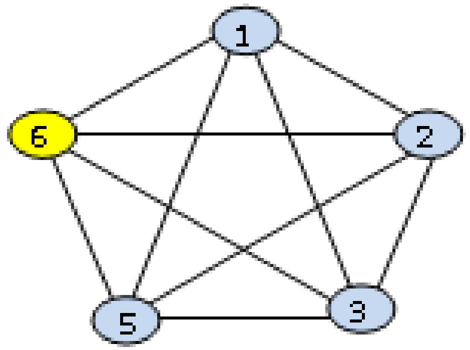
Known Result: Let G be a complete graph K_n of order n and Fibonacci sequence F_n , defined as $F_n = F_{n-1} + F_{n-2}$, for $n \geq 3$, with seed values $F_1 = 1$ and $F_2 = 2$ then:

$$es(K_n) \leq F_n$$

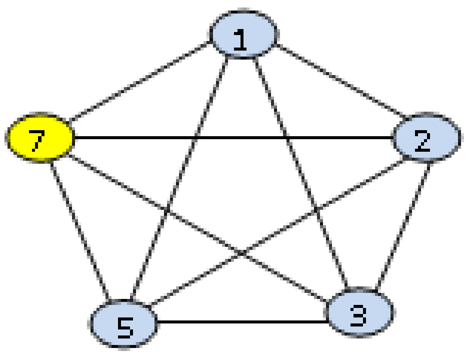
A. Ahmad, O. Al-Mushayt, M. Baca, On edge irregularity strength of graphs, Appl. Math. Comput. 243(2014), 607–610.

Algorithm for K_5 Adjacency Matrix

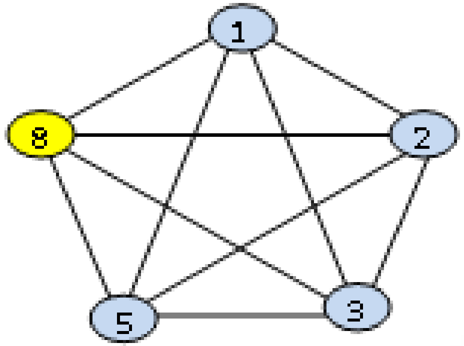
	1	2	3	5	6
1	0	3	4	6	7
2		0	5	7	8
3			0	8	9
5				0	11
6					0



	1	2	3	5	7
1	0	3	4	6	8
2		0	5	7	9
3			0	8	10
5				0	12
6					0

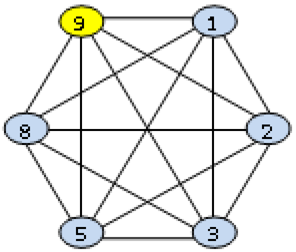


	1	2	3	5	8
1	0	3	4	6	9
2		0	5	7	10
3			0	8	11
5				0	13
6					0

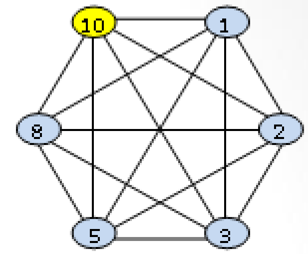


Algorithm for K_6 Adjacency Matrix

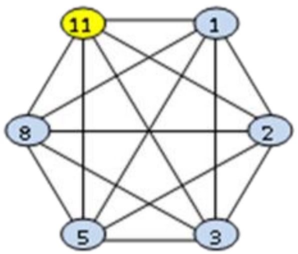
	1	2	3	5	8	9
1	0	3	4	6	9	**10**
2		0	5	7	**10**	11
3			0	8	11	12
5				0	13	14
6					0	15
9						0



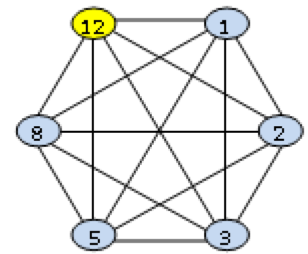
	1	2	3	5	8	10
1	0	3	4	6	9	**11**
2		0	5	7	10	12
3			0	8	**11**	13
5				0	13	15
6					0	16
9						0



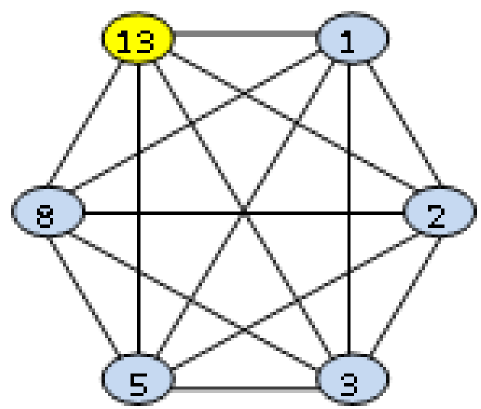
	1	2	3	5	8	11
1	0	3	4	6	9	12
2		0	5	7	10	**13**
3			0	8	11	14
5				0	**13**	16
6					0	17
9						0



	1	2	3	5	8	12
1	0	3	4	6	9	**13**
2		0	5	7	10	14
3			0	8	11	15
5				0	**13**	17
6					0	18
9						0



	1	2	3	5	8	13
1	0	3	4	6	9	14
2		0	5	7	10	15
3			0	8	11	16
5				0	13	18
6					0	19
9						0

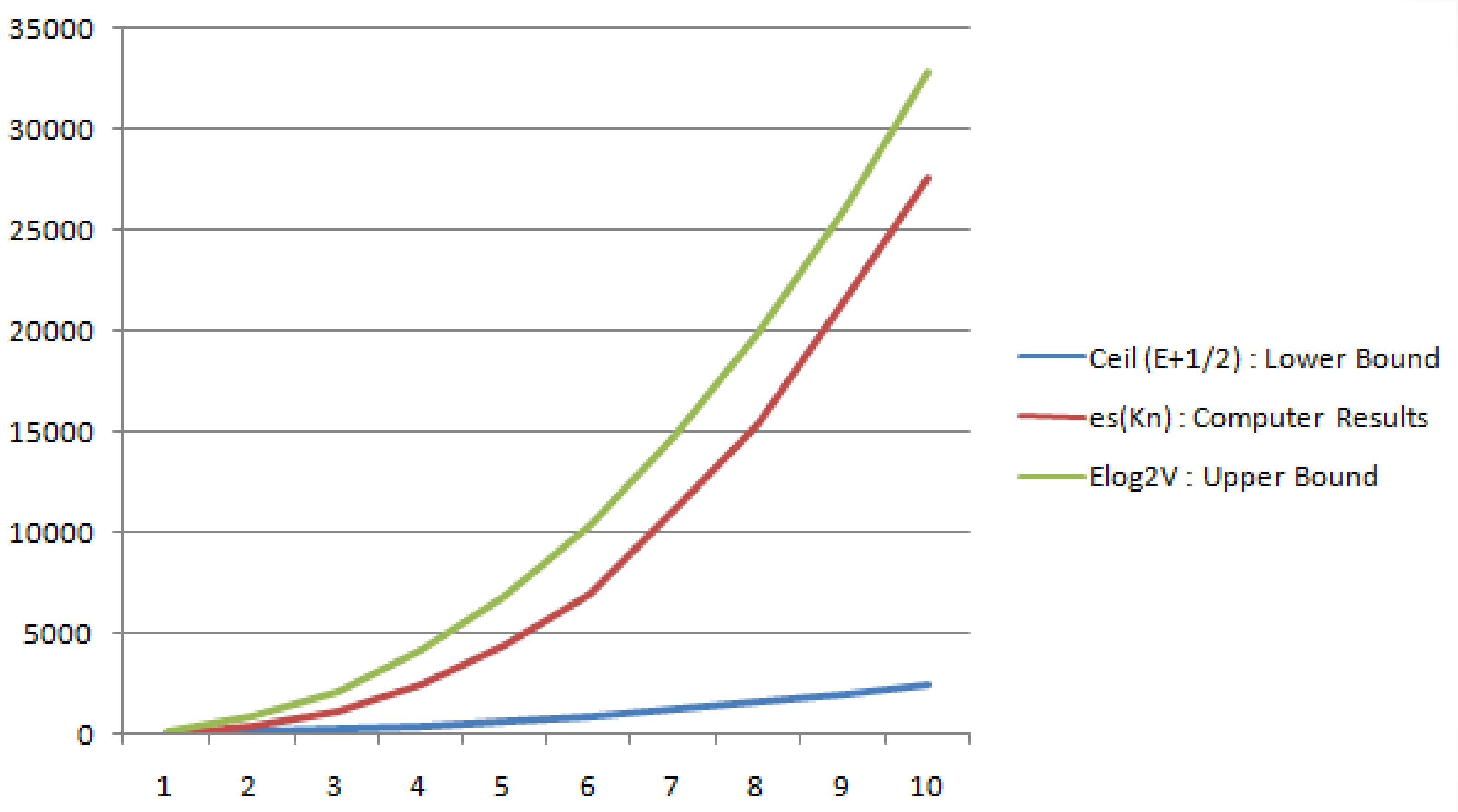


Improved upper bound for es(K_n)

V	E	es(K_n) : Computer Results	es(K_n) $\leq E\log_2 V$	F_n
10	45	53	149	55
20	190	413	821	6765
30	435	1161	2134	832040
40	780	2497	4151	102334155
50	1225	4447	6913	12586269025
60	1770	6980	10455	1548008755920
70	2415	11110	14802	190392490709135
80	3160	15470	19977	23416728348467600
90	4005	21492	25999	2880067194370810000
100	4950	27602	32887	354224848179261000000

Asim, M. A., Ahmad, A. and Hasni, R., Iterative algorithm for computing irregularity strength of complete graph, Ars Combin. 138 (2018), 17-24

Improved upper bound for es(K_n)



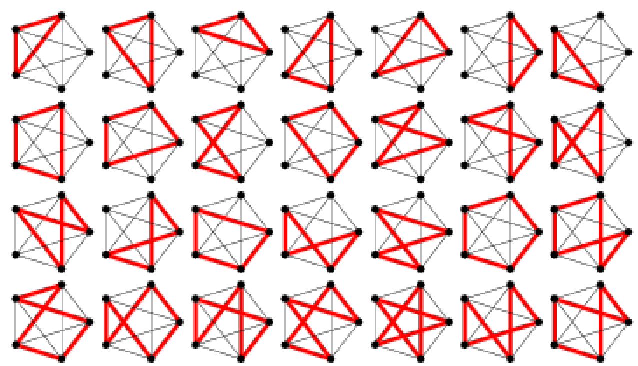
International Conference on Graph Theory and Information Security
ICGTIS-2017 Indonesia.

Prof. Baca, accepted the power of Algorithmic results.

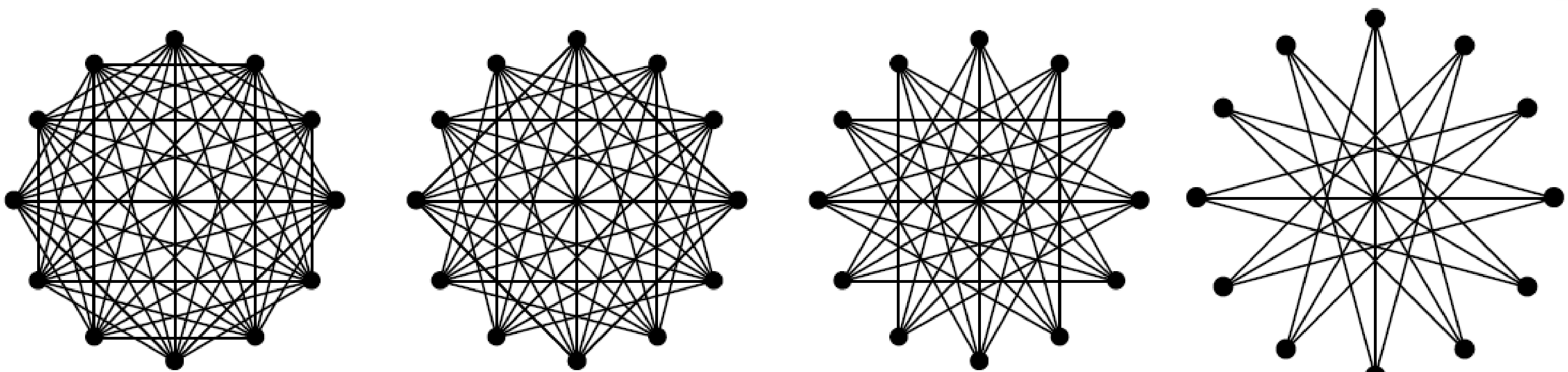
$E\log_2 V << F_n$

Graph De-composition

By deleting edges from complete graph, sub-graphs can be extracted like paths, cycles, star graphs and disjoint graphs.



More interested types of ***r-Regular graphs*** can be extracted by deleting Hamiltonian cycles or *j*-factors from complete graph.

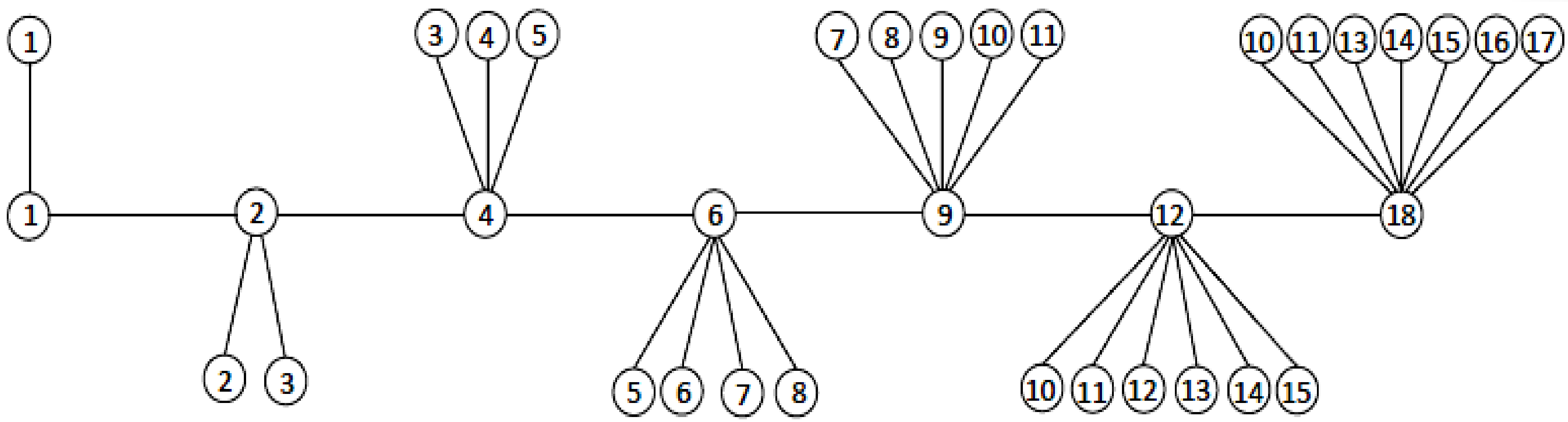


9-regular Graph : $G_{12,9}$ 7-regular Graph : $G_{12,7}$ 5-regular Graph : $G_{12,5}$ 3-regular Graph : $G_{12,3}$

M. A. Asim, R. Hasni, A. Ahmad, B. Assiri, A. S. Fenovcikova, “Irregularity Strength of Circulant Graphs using Algorithmic Approach”, IEEE Access, 2021.

Non - Homogeneous Caterpillar

- $T = CT_n(m_1,\ m_2,\ \ldots m_n)$ a non-homogeneous caterpillar with $m_{i+1} = m_i + 1$ for $1 \le i \le n-1$ with $m_1 = 1$.
- Order of T is ***n(n+3)/2***.

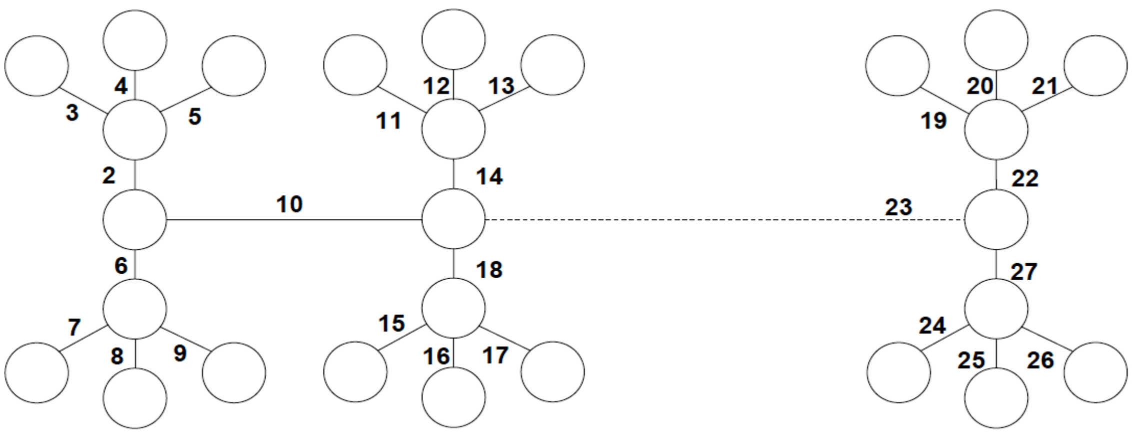


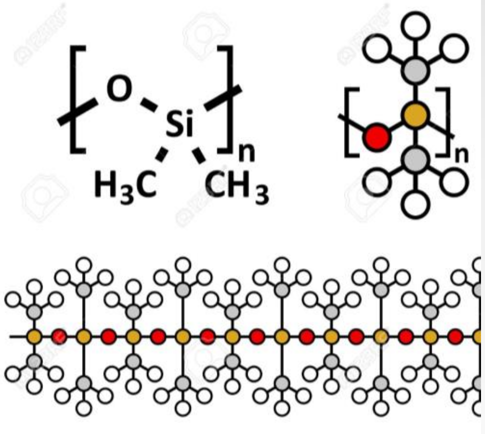
Vertex label is at most ***k*** and the edge weights are unique

thus T admits edge irregular $k = \left\lceil \dfrac{\left| \dfrac{n(n+3)}{2} \right|}{2} \right\rceil - labeling$

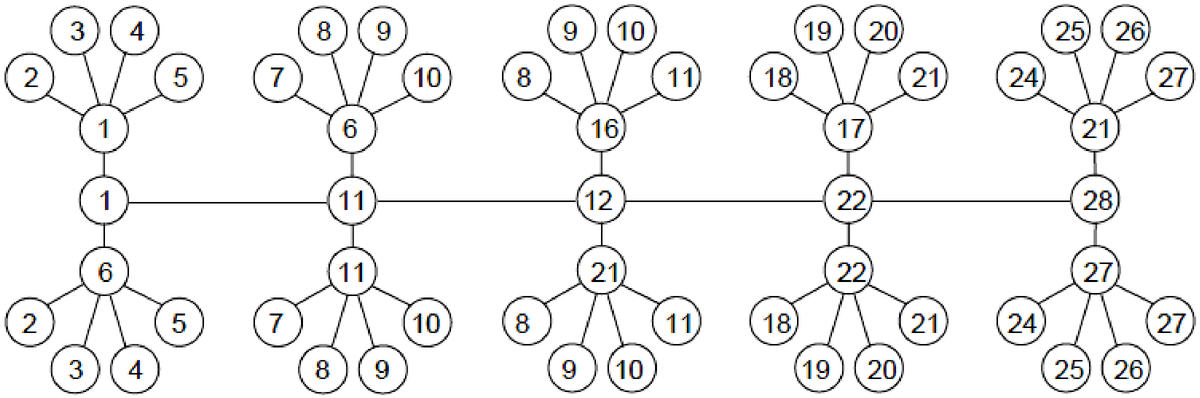
Homogeneous Lobster

- For any star graph of order ***p,*** where ***p* ≥ *2.***
- Internal vertices of two $\boldsymbol{S_p}$ are connected with each vertex of path graph $\boldsymbol{P_n}$.
- New structure will become lobster ***Lob(n, p)***.
- Order o





Homogeneous Lobster



All vertex labels are at most ***k*** and the edge weights are distinctive, thus ***Lob(n,p)*** admits the edge irregular *k*-labeling.

$$\boldsymbol{k} = \left\lceil \frac{|\boldsymbol{n}(\mathbf{2}\boldsymbol{p} + \mathbf{1})|}{\mathbf{2}} \right\rceil$$

Assignment-3:

Vertex *k*-Labeling of Amalgamated Star Graphs using Algorithmic Approach

Submission Date : April 12th, 2024

Instructions:

In this assignment, 5 to 7 students can be grouped together and submit one document.

Late and copied assignment won't be graded and will get ZERO credit.

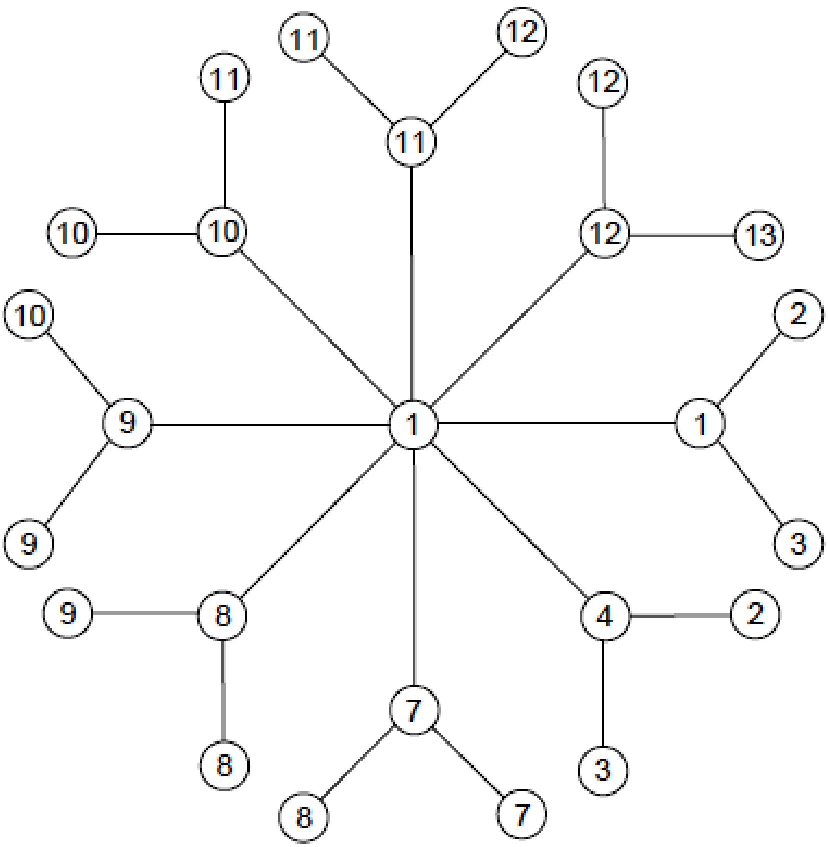
Problem-1: Homogenous amalgamated Star : $S_{n,3}$ (5 Marks)

- For $\boldsymbol{n \geq 3}$, homogenous amalgamated star $\boldsymbol{S_{n,3}}$ admits the edge irregular k-labeling.

- Order of $S_{n,3}$ = 3n+1.

- Vertex label is at most $\boldsymbol{k}$ and the edge weights are distinctive, thus $\boldsymbol{S_{n,3}}$ admits the edge irregular k-labeling.

$$\boldsymbol{k = \left\lceil \frac{3n+1}{2} \right\rceil}$$



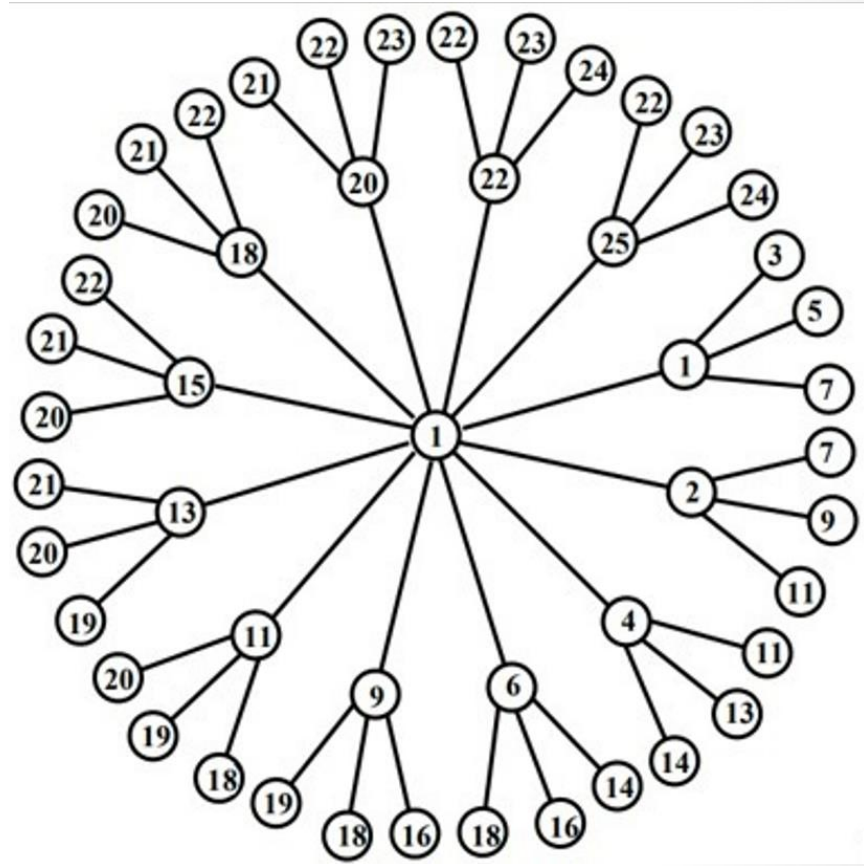
Problem-2: Homogenous amalgamated Star : $S_{n,m}$ (3 Marks)

- For $n \geq 3$, homogenous amalgamated star $S_{n,m}$ admits the edge irregular k-labeling.

- Order of $S_{n,m}$ = m x n + 1

- Vertex label is at most k and the edge weights are distinctive, thus $S_{n,m}$ admits the edge irregular k-labeling.

$$k = \left\lceil \frac{m \times n + 1}{2} \right\rceil$$



Assig-3: Tasks to Do

1. Find out the best data-structure to represent / store the graph in memory.
2. Devise an algorithm to assign the labels to the vertices using vertex k-labeling definition. (Main Task)
3. What design strategy you will apply, also give justifications that selected strategy is most appropriate.
4. How traversing will be applied?
5. Store the labels of vertices and weights of the edges as an outcome.
6. Compare your results with mathematical property and tabulate the outcomes for comparison.
7. Hardware resources supported until what maximum value of n, m.
8. Compute the Time Complexity of your algorithm $T(V,E)$ or $T(n)$.

Problem-3: For any number n as branches with centroid vertex. (2 Marks)

Tasks to Do:

1. Suggest a suitable name.
2. Devise the formulae for calculating order and size of the graph.
3. Data-structure to store the graph.
4. Assign the labels using algorithm.
5. Store the labels of vertices and weights of the edges as an outcome.