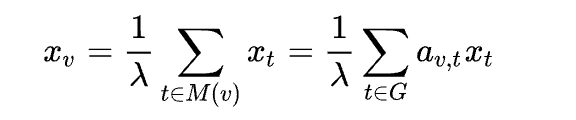
**EIGEN VECTOR CENTRALITY**

* This metric measures the importance of a node in a graph as a function of the importance of its neighbours. If a node is connected to highly important nodes, it will have a higher Eigen Vector Centrality score as compared to a node which is connected to lesser important nodes.
* EVC is its short abbreviation.
* It is the measure of the influence of a node in a network.
* In a network, nodes are connected by edges, and Eigen vector centrality aims to quantify the influence or prominence of a node based on its connections and the centrality of its neighbouring nodes. The underlying assumption is that a node is important if it is connected to other important nodes.
* It assigns relative scores to all the nodes in the network based on the concept that the a node would contribute more to its score if it is connected to high scoring nodes than low scoring nodes.
* We can calculate the EVC of a node from the adjacency matrix of the given network of things.
* —-> For a Graph G(V,E), where |V|: number of vertices in the graph network

A =( a v, t) be the adjacency matrix of the network.

( a v, t)=1 ,if the vertex ‘v’ is directly linked to vertex ‘t’

( a v, t)=0, if the vertex ‘v’ is not directly linked to vertex ‘t’

The relative centrality score of the vertex v can be defined as :

Where M(v) is a set of neighbours of

λ: a constant.

From a small rearrangement, i.e when we multiply both LHS and RHS of the

expression with λ ,then we realise that new RHS part is which is

nothing but the product of the adjacency matrix A with the vector v. i.e Ax.

The LHS part is product of vector x with the scalar constant λ i.e xλ.

So, with a small rearrangement, this expression can be rewritten in the vector

as the Eigen vector equation **Ax=** λ**x**.

—-> Generally, there exist many values of Eigen value λ for which non-zero Eigen

vector solutions exist.

But,according to the **Perron-Frobenius theorem** shows us how a

particular Eigen value should be chosen according to the Adjacency

matrix.

—->Specifically, for a nonnegative square matrix (such as an adjacency matrix) with entries greater than or equal to zero, the Perron-Frobenius theorem states that there exists a nonnegative real eigenvalue called the Perron eigenvalue, denoted as λ\_max, which is greater than or equal to the absolute value of all other eigenvalues.

Furthermore, the Perron eigenvalue λ\_max has a corresponding nonnegative eigenvector called the Perron eigenvector, denoted as v, which has strictly positive entries.  
In the context of an adjacency matrix, which represents the connections between nodes in a graph, the Perron eigenvalue and the corresponding Perron eigenvector have important interpretations.

The Perron eigenvalue λ\_max represents the dominant eigenvalue of the adjacency matrix and provides information about the connectivity and structure of the graph. Its magnitude indicates the rate of growth of a random walk on the graph, and it determines the stability and behavior of various dynamic processes on the graph.

* General Process of Calculating the EVC of a node in the network at any iteration is as follows:
* Find the Adjacency matrix A of the given node network.
* Before the iteration, find the degree of each node(number of nodes to which that particular node is directly connected with) and represent the entire degrees collectively in a 1D column vector .
* Now, the resultant 1D column vector is given by the product of A with .

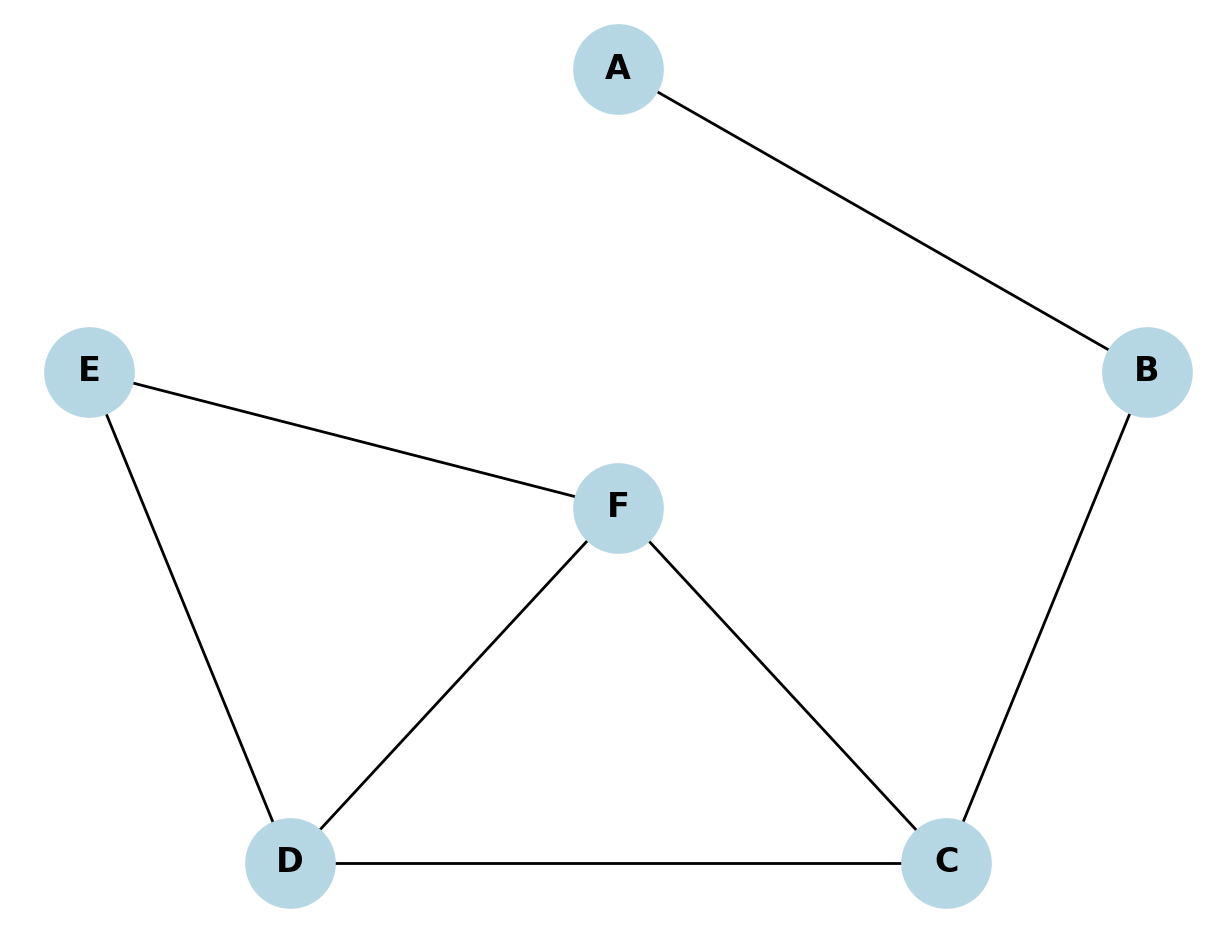
=A

Each element in the resultant vector represents the centrality scores of the nodes in the network after that particular iteration.

* To calculate the absolute score normalise the Eigen Vector(divide that particular score with the sum of other scores in the vector such that the total sum of absolute is 1).
* The calculated EVC score of the network shows its strength after several iterations.Repeated multiplication makes the EVC score of every node to eventually be a function of or dependent on several degrees of its neighbouring nodes, thereby providing a globally accurate EVC score for each node.Usually the process of multiplying the EVC vector with the adjacency matrix is repeated until the EVC values for nodes in the graph reach an equilibrium or stop showing appreciable change.

**Taking an Example to explain the concept of Calculating Eigen Vector Centrality of the Nodes in a given network:**

This is a graph representing a network :



* The Adjacency matrix A of the above graph is
* A=
* Step2: Finding the degree of each node and representing it in the form of a column vector:

| Node | Degree |
| --- | --- |
| A | 1 |
| B | 2 |
| C | 3 |
| D | 3 |
| E | 2 |
| F | 3 |

(number of nodes to which that particular node is directly connected with)

Now, representing the degree of all nodes initially before any iteration in the form of a vector.

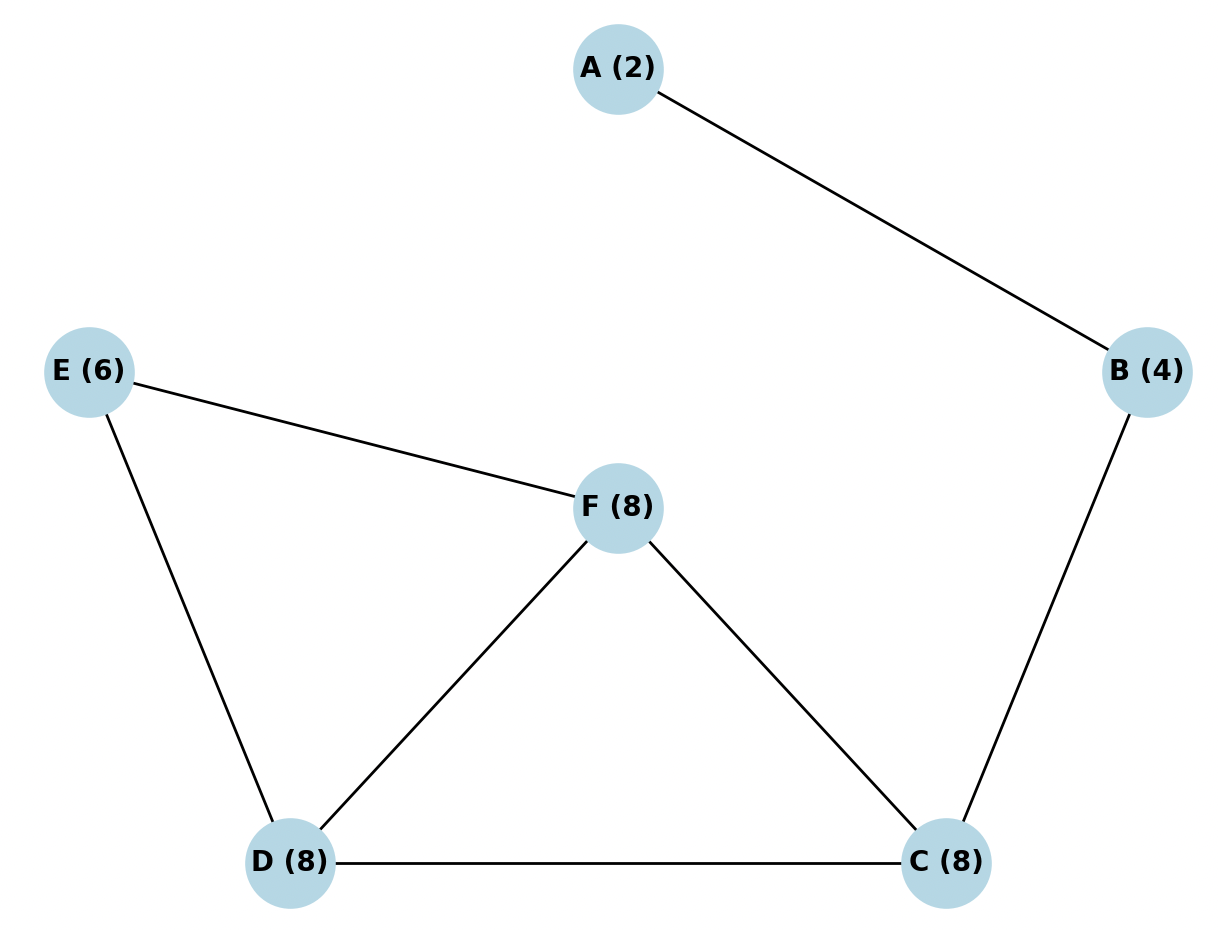
V==

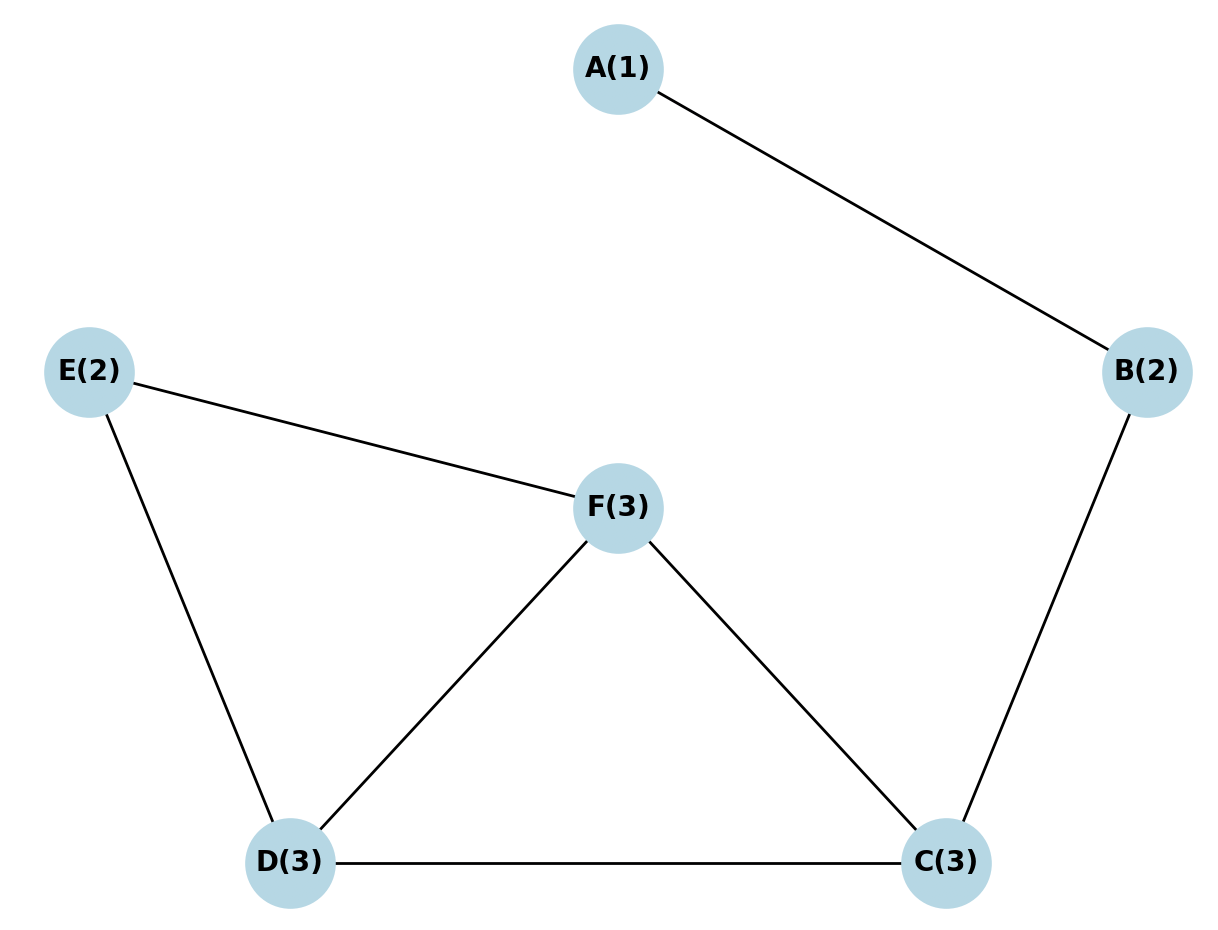
* Step 3:Calculating mathematically the Eigen vector Centrality of each node

After first iteration:

=

==

The effect of first iteration of multiplication can be visualised as shown below:



—>

As we see above the nodes C,D,E have highest scores of 8 since these nodes are connected to multiple nodes(3) each with high degrees(importance)  
But, the node A has a least score of only 2 as it was connected only with a single node B   
-> We can observe that the degree of a node after iteration is the sum of the degrees of the nodes to which it is connected to before the iteration.

For example the degree of F after iteration is 8.which is

degree(E)+degree(D)+degree(C)

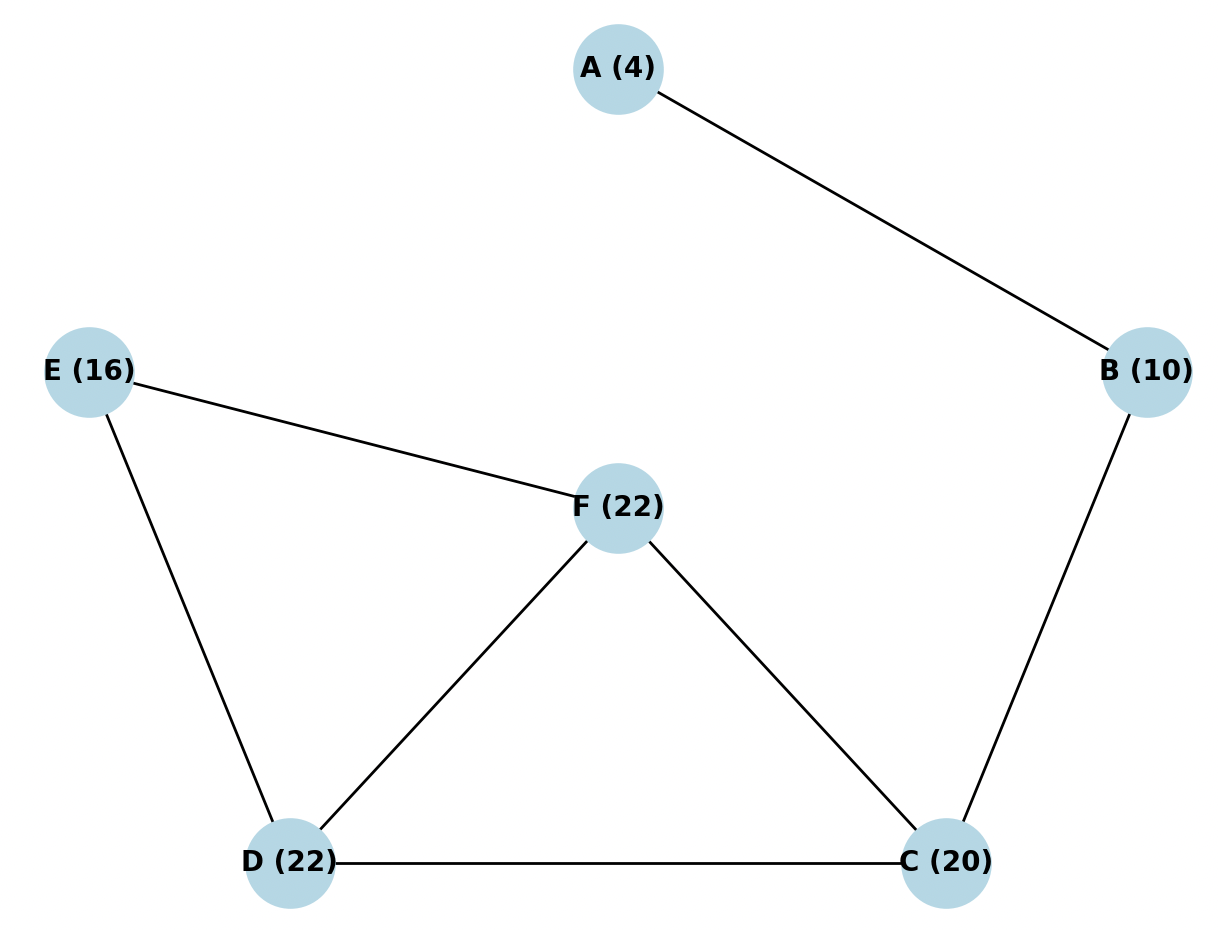
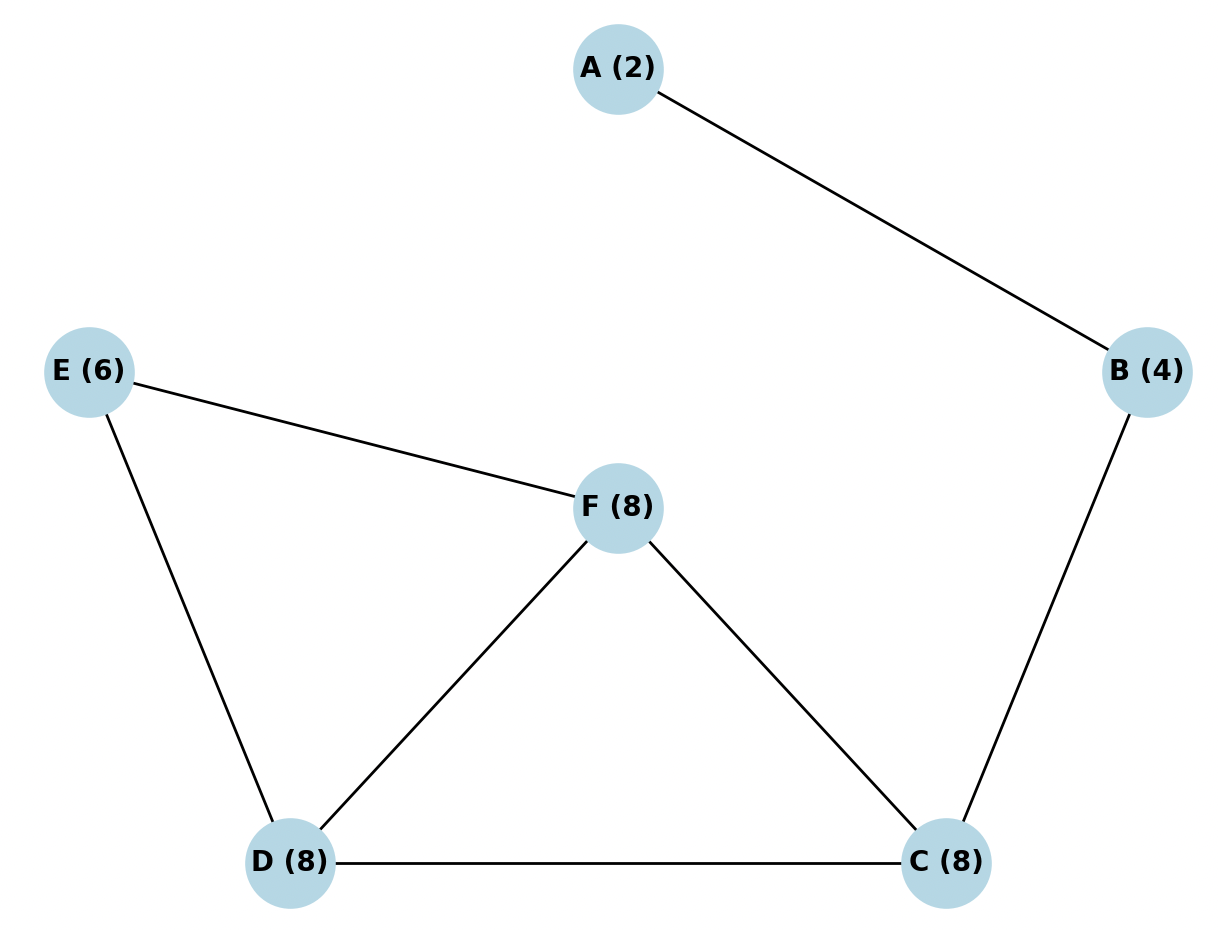
=2+3+3=8

* Now, repeating the same process to calculate the EVC of each node after 2nd iteration and observe the changes happening:

=

=

==

So, now the effect of multiplication after the 2nd iteration can be visualised as:

->

* So, here we are multiplying the EVC vector again and again with the adjacency matrix. the answer to that lies in the fact that multiplying the resultant vector again with the adjacency matrix of the graph helps the EVC score spread out in the graph so as to get a more globally prominent EVC score vs a localised EVC score for each node in the graph. If we observe, after the first iteration of multiplication, each node’s EVC score is a function of only its direct (1st degree) neighbours, thus is a localised score which might not be accurate at a global level in the graph
* Elaborating the above operation in brief:  
  ->After the first iteration of multiplication, each node gets it’s EVC score from its direct(1st degree) neighbours.  
    
  ->In the second iteration, when we multiply the resultant vector again with the adjacency matrix, each node again gets it’s EVC score from its direct neighbours but the difference in the second iteration is that this time, the scores of the direct neighbours have already been impacted by their own direct(1st degree) neighbours previously(from the first iteration of multiplication) which eventually helps the EVC score of any node to be a function of its 2nd degree neighbouring nodes as well.  
    
  -> In subsequent iterations of multiplication, the EVC score of graph nodes keeps getting updated by getting impacted by EVC scores from neighbouring nodes of farther degree (3rd, 4th and so on).

-> Repeated multiplication makes the EVC score of every node to eventually be a function of or dependent on several degrees of its neighbouring nodes, thereby providing a globally accurate EVC score for each node.Usually the process of multiplying the EVC vector with the adjacency matrix is repeated until the EVC values for nodes in the graph reach an equilibrium or stop showing appreciable change.

* If we normalise the vector and then calculate the subsequent normalised EVC vectors, after certain iterations the the EVC values of the nodes would saturate and do not undergo a further change for further iteration.

—> That EVC vector is the Eigen Vector for that particular graph node network.

—-> **Ax= λx** is satisfied according to the property of Eigen vectors.

**Application:**

Finally,

Eigenvector centrality is calculated using the concept of eigenvectors and eigenvalues in linear algebra.

* **Google's PageRank algorithm** uses eigenvector centrality to rank web pages in order of their importance. The idea behind PageRank is that a web page is considered important if it is linked to by other important web pages. It treats the web as a directed graph, where each web page is a node and the links between them are the edges.  
    
  The PageRank algorithm starts by assigning an initial score to each web page. These scores are then updated iteratively based on the scores of the pages that link to them. The update process involves calculating the eigenvector centrality of each page, considering both the incoming and outgoing links.  
    
  In terms of linear algebra, the PageRank algorithm represents the web graph as a stochastic matrix, typically referred to as the transition matrix. This matrix describes the probability of transitioning from one page to another through a link. The PageRank scores of the pages are obtained by finding the principal eigenvector of this transition matrix.  
    
  The eigenvector corresponding to the principal eigenvalue represents the stationary distribution of the random walk on the web graph. The PageRank scores are essentially the elements of this eigenvector, indicating the relative importance or popularity of each page in the network.  
    
  By utilising eigenvector centrality, Google's PageRank algorithm considers the global structure of the web graph and determines the importance of a page based on its connections to other important pages. This approach helps in ranking web pages and determining their relevance for search engine results.  
    
  Overall, eigenvector centrality and the concepts from linear algebra play a crucial role in Google's PageRank algorithm by providing a mathematical framework to assess and rank the importance of web pages in a network of interconnected links.  
  ->Some other applications:
* Social Network Analysis:

Eigen vector centrality is widely used in social network analysis to identify influential individuals within a network. By analysing connections and relationships between individuals, Eigen vector centrality helps uncover key players who have a significant impact on the flow of information or influence within the network.

* Biological Networks:

Eigen vector centrality is applied in biological networks, such as gene regulatory networks or protein-protein interaction networks. It helps identify central genes or proteins that play crucial roles in biological processes, providing insights into the functional importance and potential drug targets within the network.

Eigen Vector Centrality also finds its applications in Recommended Systems, transportation networks mainly in helping the underlying structure and dynamics of complex systems.