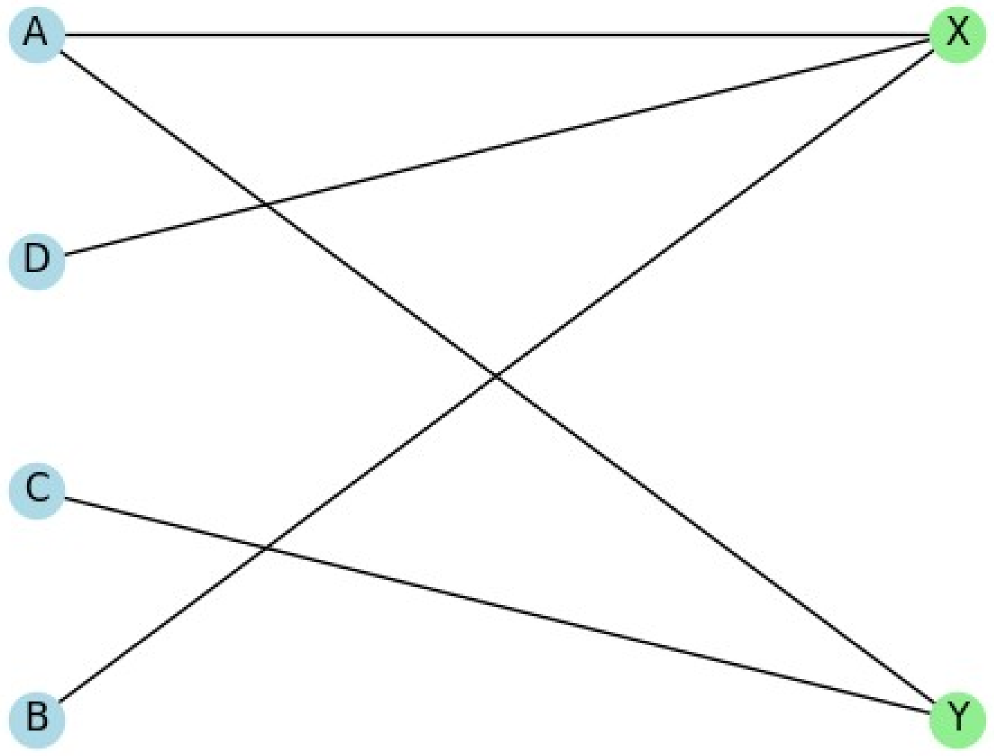
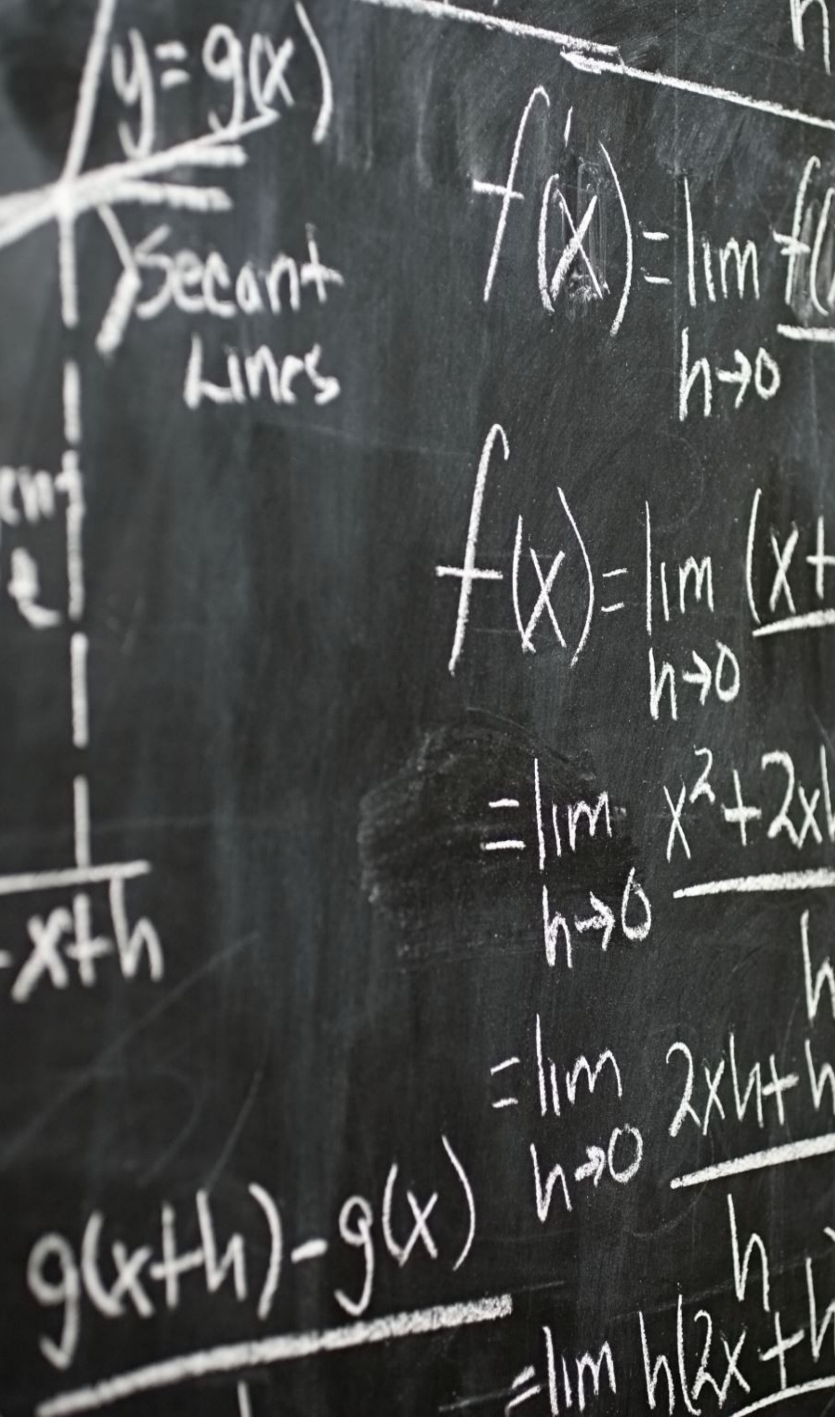
Bipartite Graphs

How does maximal optimisation for favours from two parties work?

What Are Bipartite Graphs ?





Mathematical Representation.

- Graph is Collection of Nodes and Edges

Where

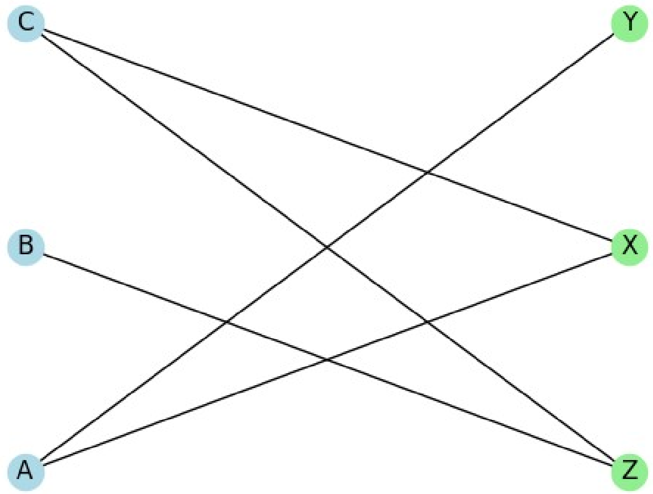
- Bipartite Graphs have two set of Nodes such that There is no edge between any two node of one set.

Mathematically :

Bipartite Graphs representation :

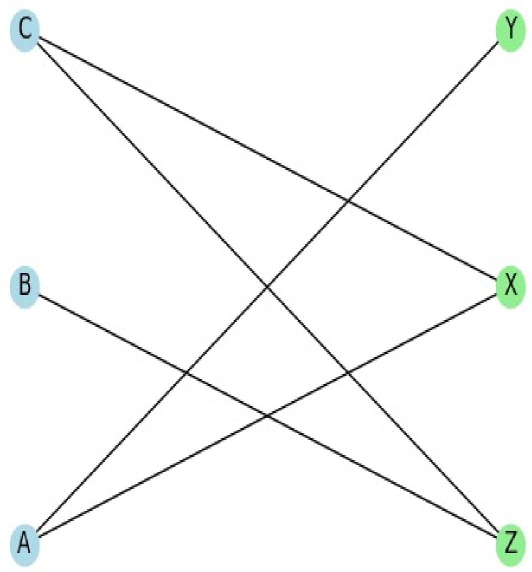
```
import networkx as nx
import matplotlib.pyplot as plt
import numpy as np

# Create an instance of the bipartite graph
B = nx.Graph()

# Define the two sets of vertices
V1 = {'A', 'B', 'C'}
V2 = {'X', 'Y', 'Z'}

# Add the vertices to the graph
B.add_nodes_from(V1, bipartite=0)  # Set bipartite=0 for the first set
B.add_nodes_from(V2, bipartite=1)  # Set bipartite=1 for the second set

# Define the edges
edges = [('A', 'X'), ('A', 'Y'), ('B', 'Z'), ('C', 'X'), ('C', 'Z')]

# Add the edges to the graph
B.add_edges_from(edges)

# Separate the vertices into two sets based on the bipartite attribute
X, Y = nx.bipartite.sets(B)

# Visualize the bipartite graph
pos = nx.bipartite_layout(B, X)

# Create a new figure for the first plot
plt.figure(1)

# Draw the vertices from each set with different colors
nx.draw_networkx_nodes(B, pos, nodelist=X, node_color='lightblue')
nx.draw_networkx_nodes(B, pos, nodelist=Y, node_color='lightgreen')

# Draw the edges
nx.draw_networkx_edges(B, pos)

# Draw labels for the vertices
nx.draw_networkx_labels(B, pos)

# Display the first plot
plt.axis('off')

adj_matrix = nx.to_numpy_matrix(B, nodelist=sorted(B.nodes()))

# Create a new figure for the second plot
plt.figure(2)

# Plot the adjacency matrix as a heatmap
plt.imshow(adj_matrix, cmap='viridis', interpolation='nearest')
plt.title('Adjacency Matrix')
```

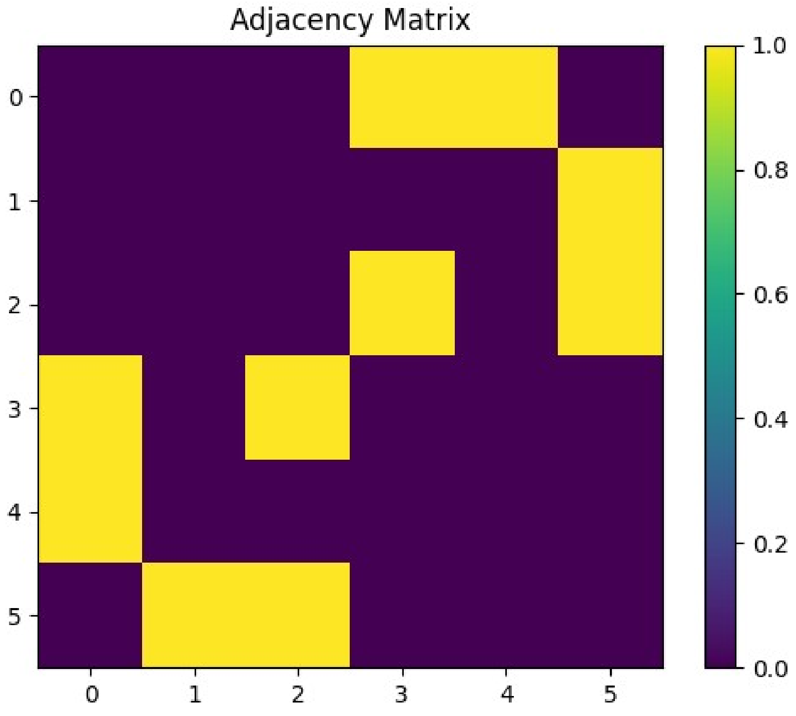
Visualisation

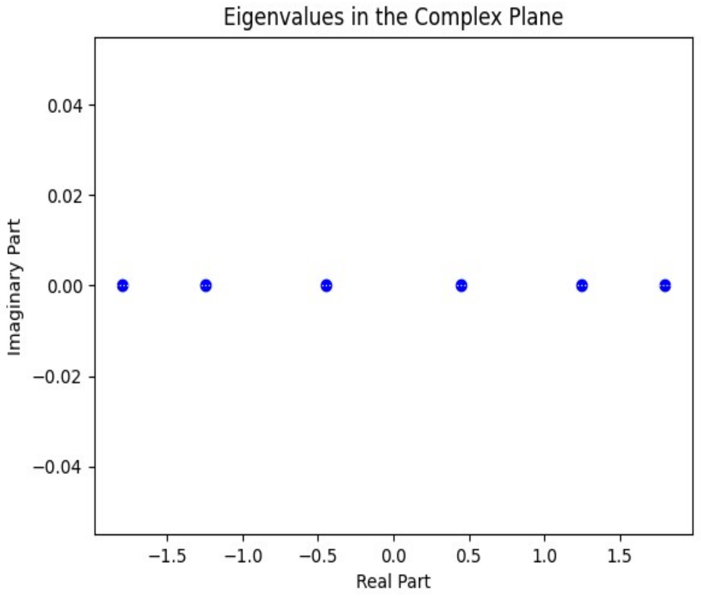


**Codes are also in submission

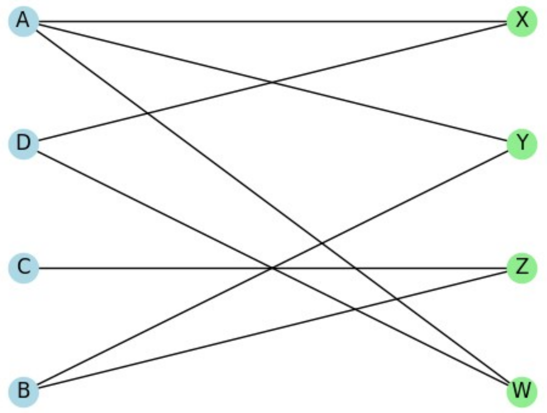
Adjoint Matrix , Graph and Eigen Plot

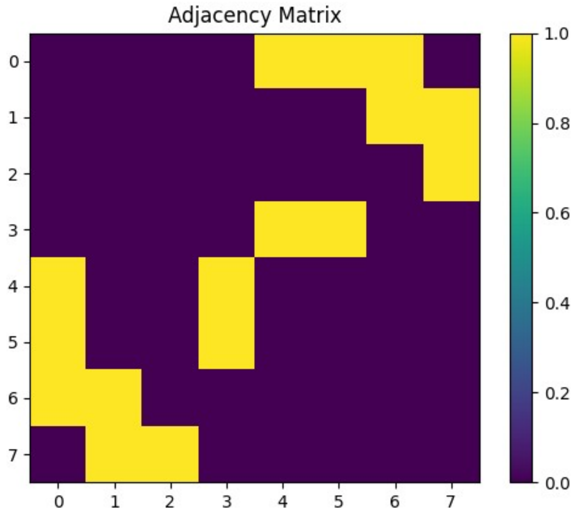


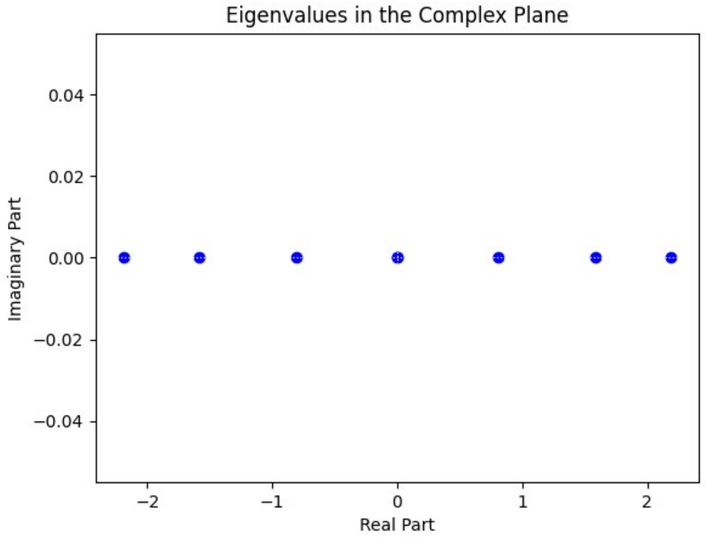




Adjoint Matrix , Graph and Eigen Plot







Applications

- Matching and Assignment Problems :

Entries from one set need to be paired with another based on certain criteria

Bipartite Graphs model such problems and device efficient algorithms to find optimal solutions.

Applications of this include Resource Allocation, Job Assignment and project Scheduling.

Biological and Social Networks

- Bipartite graphs find applications in analyzing biological networks, such as protein-protein interaction networks or gene-disease association networks.
- Similarly, bipartite graphs are utilized in social network analysis to study interactions between individuals and groups, friendship networks, and information diffusion.

Relation with Eigenvalue and Proof

Theorem 1 (Eigenvalues of Bipartite Graphs)**.** *Let* $\mathbf{A}$ *be the adjacency matrix of a bipartite graph. By definition, a bipartite graph has its vertices divided into two disjoint sets such that all edges connect vertices from different sets. The eigenvalues of* $\mathbf{A}$ *appear in pairs of* $+a$ *and* $-a$*, where* a *is a real number.*

Proof. We will prove the statement by considering the properties of the adjacency matrix of a bipartite graph.

Step 1: Matrix A^2

Consider the matrix $\mathbf{A}^2$. The (i, j)th entry of $\mathbf{A}^2$ represents the number of paths of length 2 between vertex i and vertex j. In a bipartite graph, there are no direct edges between vertices within the same set, so all entries on the diagonal of $\mathbf{A}^2$ are zero.

Step 2: Equivalence of A^2v and Av^2

For any vector $\mathbf{v}$, we have $(\mathbf{A}^2)\mathbf{v} = \mathbf{A}(\mathbf{v}^2)$. This equation can be proven using matrix multiplication.

Step 3: Eigenvalue λ and eigenvector $\mathbf{v}$

Let λ be an eigenvalue of $\mathbf{A}$ with eigenvector $\mathbf{v}$. By definition, $\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$.

Step 4: From $Av = \lambda v$ to $Av^2 = \lambda(Av)$

Multiply both sides of the equation by $\mathbf{A}$: $\mathbf{A}(\mathbf{Av}) = \mathbf{A}(\lambda\mathbf{v})$. This yields $\mathbf{A}^2\mathbf{v} = \lambda(\mathbf{Av})$.

Step 5: Substituting Step 2

By substituting Step 2, we have $\mathbf{Av}^2 = \lambda(\mathbf{Av})$. This implies that λ is an eigenvalue of $\mathbf{A}^2$ with eigenvector $\mathbf{v}^2$.

Step 6: Non-negativity of A^2

The eigenvalues of $\mathbf{A}^2$ are non-negative. This is because $\mathbf{A}^2$ is a non-negative matrix representing the number of paths of length 2 in the bipartite graph.

Step 7: Eigenvalue λ is non-negative

Since λ is an eigenvalue of $\mathbf{A}^2$, we have $\lambda \geq 0$.

Step 8: Assumption: $\lambda \neq 0$

Let's assume $\lambda \neq 0$. If $\lambda \neq 0$, then the eigenvector $\mathbf{v}^2$ is nonzero.

Step 9: From $Av = \lambda v$ to $\lambda = \frac{v^T A^T v}{v^T v}$

By the definition of the eigenvalue, we have $\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$. Multiplying both sides of the equation by $\mathbf{v}^T$, we get $(\mathbf{v}^T\mathbf{A}\mathbf{v}) = \lambda(\mathbf{v}^T\mathbf{v})$.

Step 10: Symmetric property

Since $\mathbf{A}^T = \mathbf{A}$ for a bipartite graph, we have $\mathbf{A}^T\mathbf{v} = \mathbf{A}\mathbf{v}$. This implies that $(\mathbf{v}^T\mathbf{A}^T\mathbf{v}) = (\mathbf{v}^T\mathbf{A}\mathbf{v})$.

Step 11: Combining Step 9 and Step 10

Combining steps 9 and 10, we get $\lambda = \frac{v^T A^T v}{v^T v}$.

Step 12: Eigenvalue λ is real

Since $\lambda = \frac{v^T A^T v}{v^T v}$, it follows that λ is a real number.

Step 13: Conclusion: Eigenvalues appear in pairs

Therefore, we conclude that the eigenvalues of $\mathbf{A}$ appear in pairs of $+a$ and $-a$, where a is a real number.

$\square$

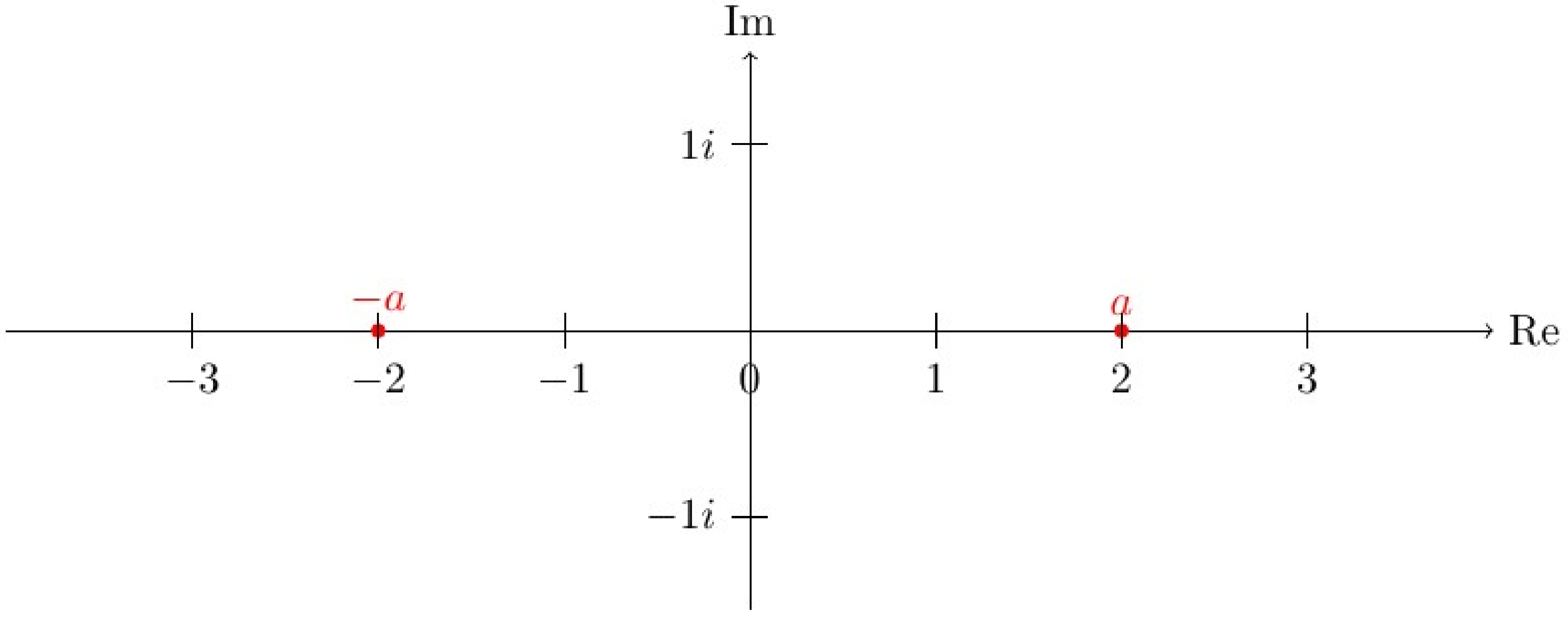


Figure 3: Eigenvalues of a Bipartite Graph