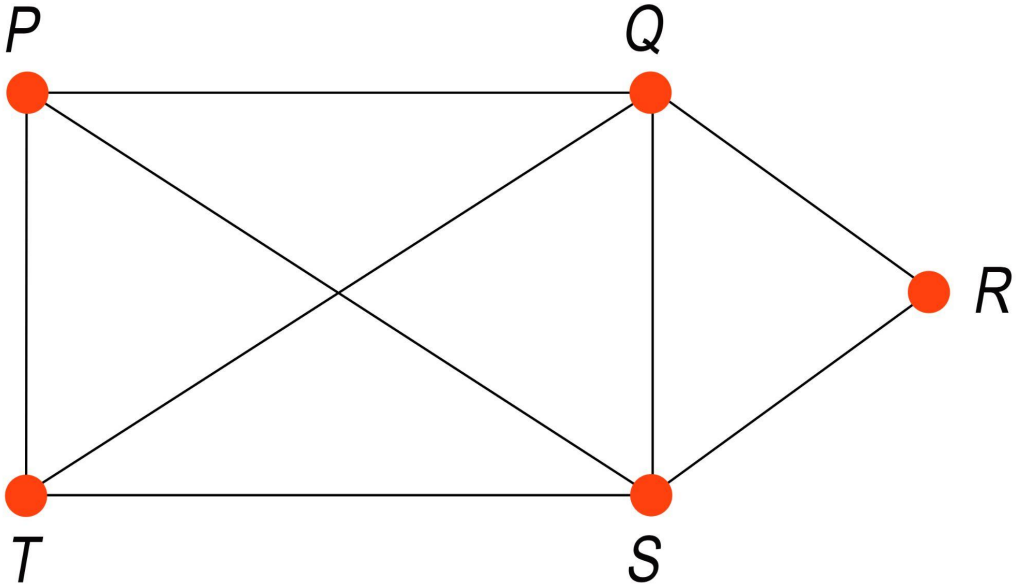
The Independence Coloring Game

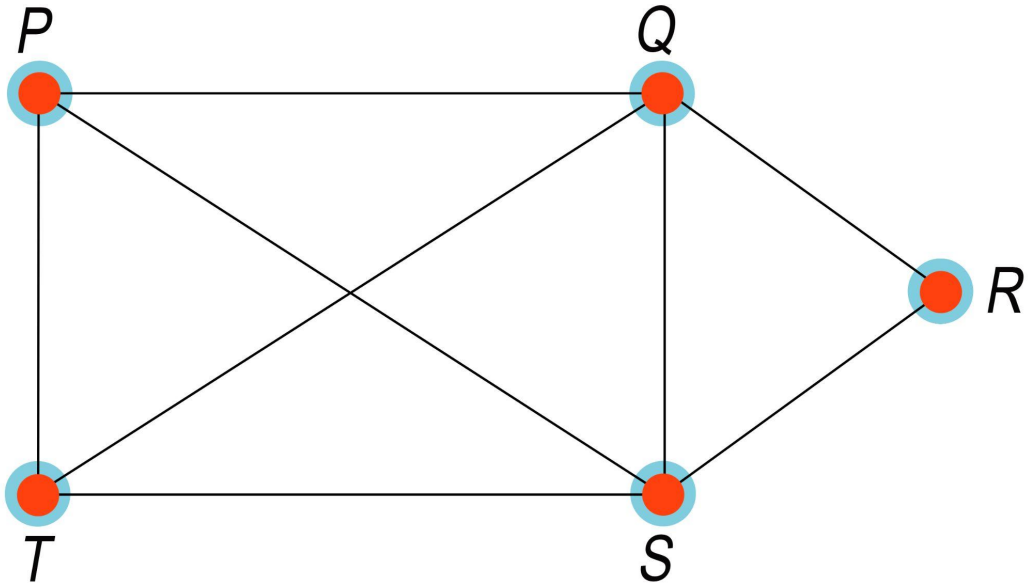
What is a graph?

A graph is composed of points, called vertices, and lines, called edges. The collection of vertices and edges is called a graph.



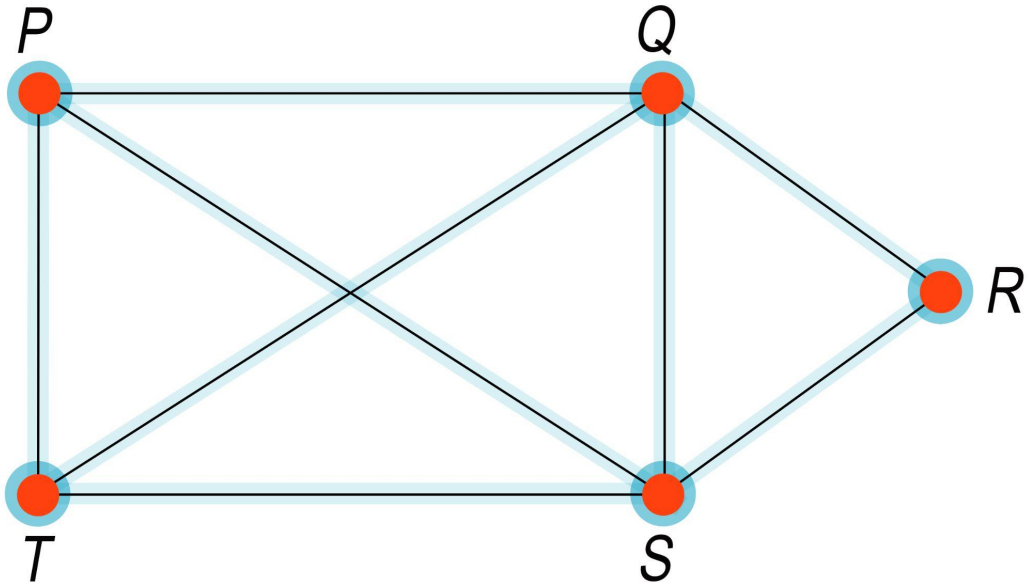
What is a graph?

A graph is composed of points, called **vertices**, and lines, called edges. The collection of vertices and edges is called a graph.



What is a graph?

A graph is composed of points, called vertices, and lines, called **edges**. The collection of vertices and edges is called a **graph**.



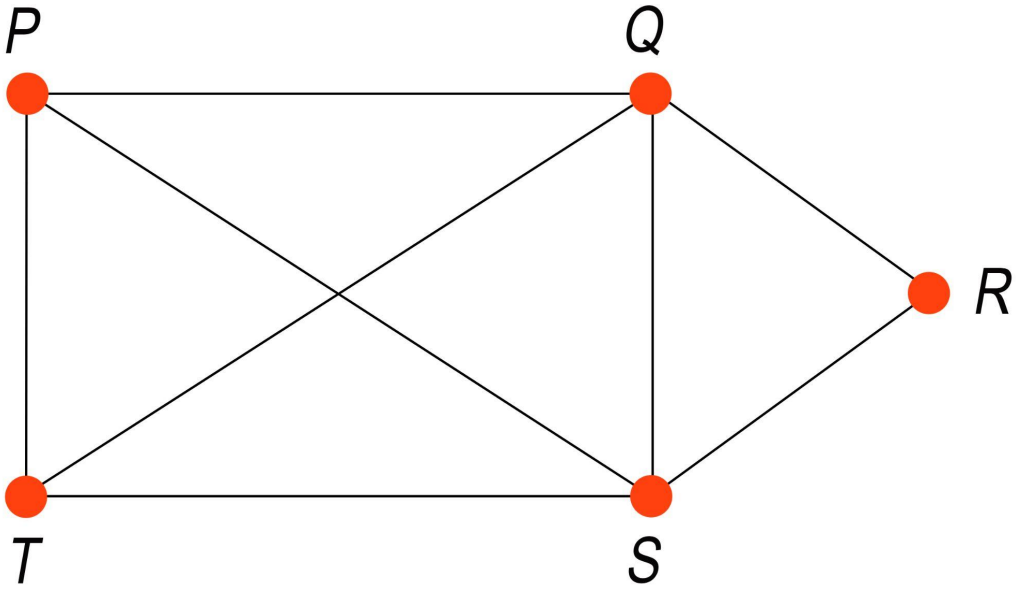
What is graph coloring?

What is graph coloring?

Graph coloring is the process of selecting and labeling vertices through coloring. There are many forms of graph coloring, but we will focus on **regular coloring** (no two adjacent vertices will be colored in the final graph).

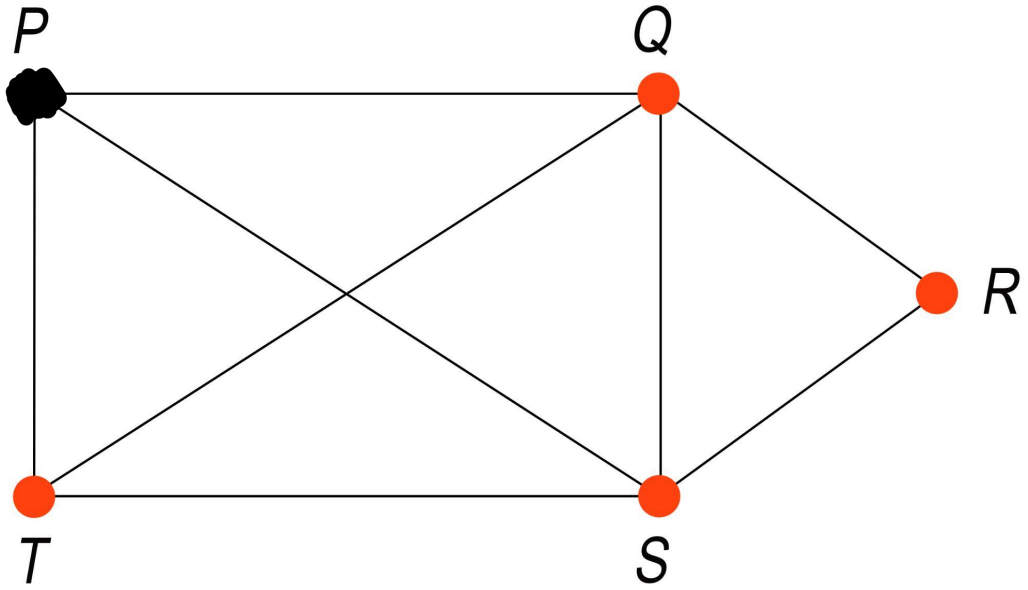
What is graph coloring?

Graph coloring is the process of selecting and labeling vertices through coloring. There are many forms of graph coloring, but we will focus on **regular coloring** (no two adjacent vertices will be colored in the final graph).



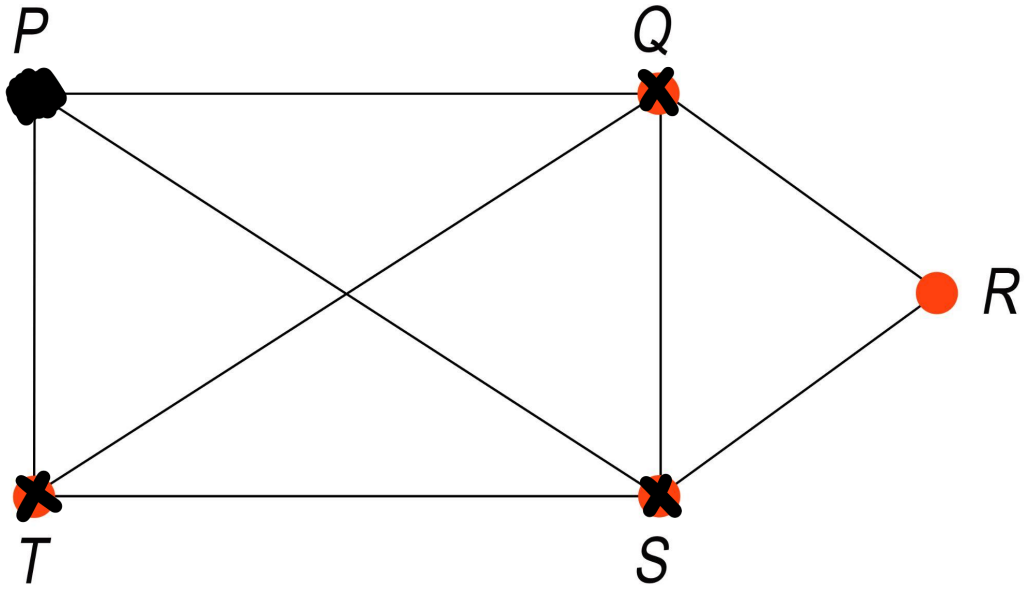
What is graph coloring?

Graph coloring is the process of selecting and labeling vertices through coloring. There are many forms of graph coloring, but we will focus on **regular coloring** (no two adjacent vertices will be colored in the final graph).



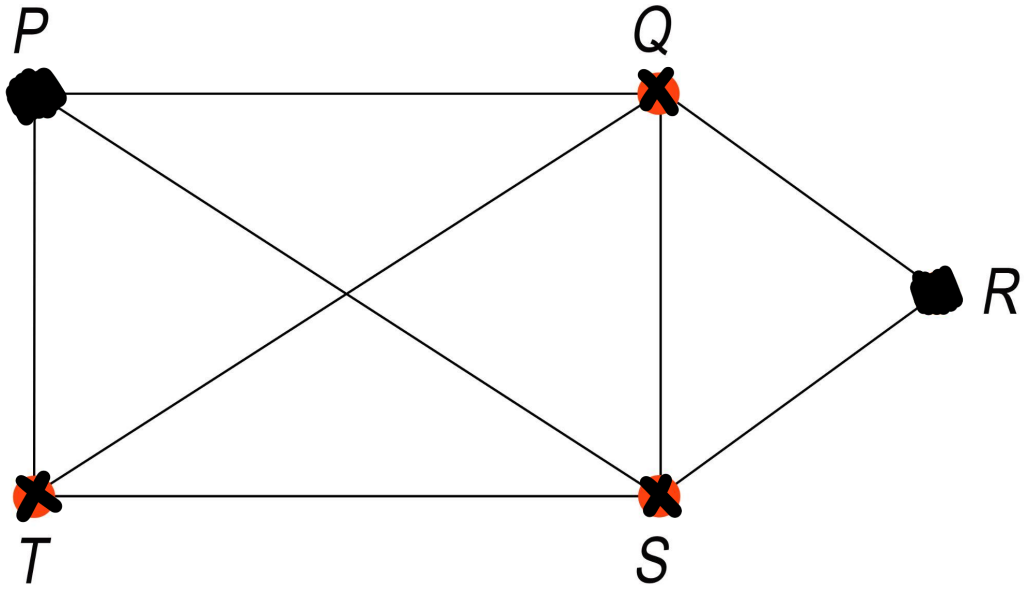
What is graph coloring?

Graph coloring is the process of selecting and labeling vertices through coloring. There are many forms of graph coloring, but we will focus on **regular coloring** (no two adjacent vertices will be colored in the final graph).



What is graph coloring?

Graph coloring is the process of selecting and labeling vertices through coloring. There are many forms of graph coloring, but we will focus on **regular coloring** (no two adjacent vertices will be colored in the final graph).



The Independence Coloring Game

The Independence Coloring Game

Minimizer's goal: Minimize the final number of colored vertices.

The Independence Coloring Game

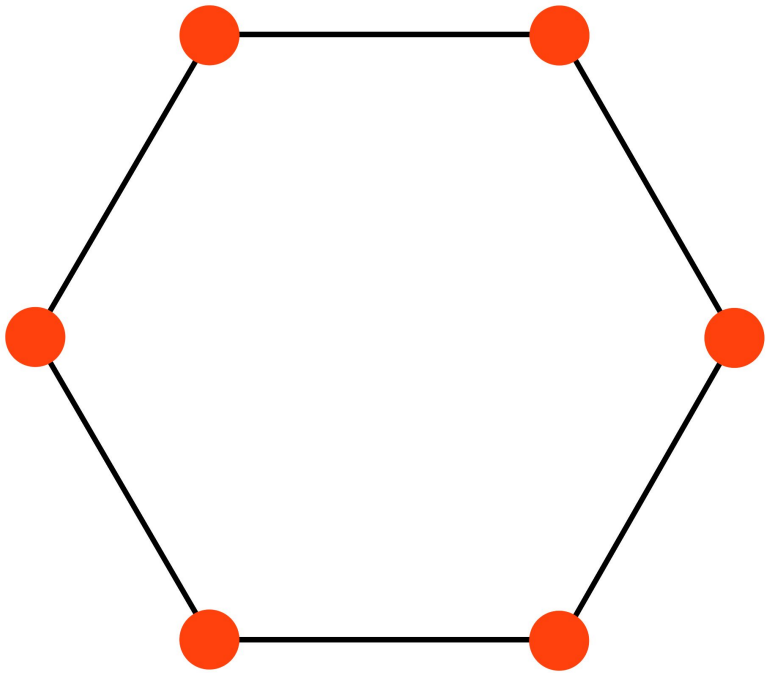
Minimizer's goal: Minimize the final number of colored vertices.

Maximizer's goal: Maximize the final number of colored vertices.

The Independence Coloring Game

Minimizer's goal: Minimize the final number of colored vertices.

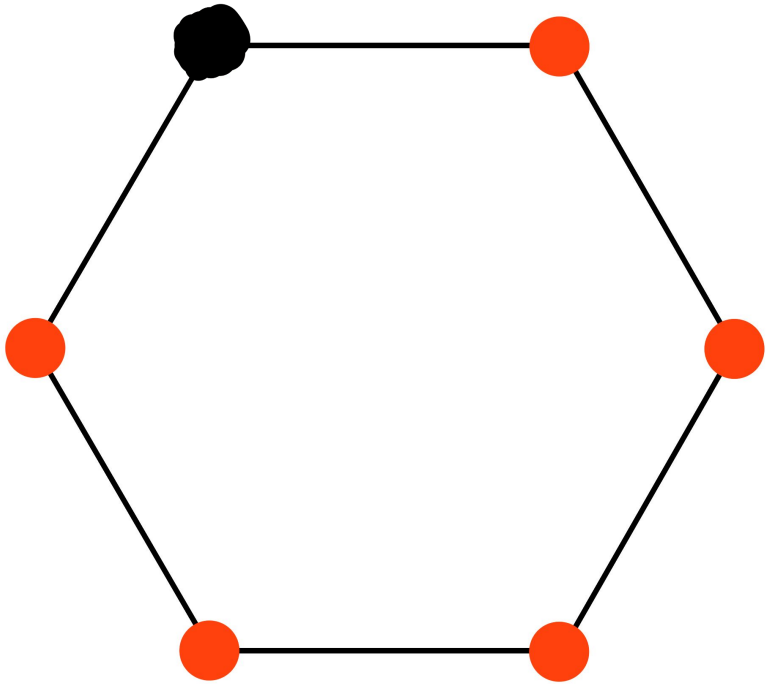
Maximizer's goal: Maximize the final number of colored vertices.



The Independence Coloring Game

Minimizer's goal: Minimize the final number of colored vertices.

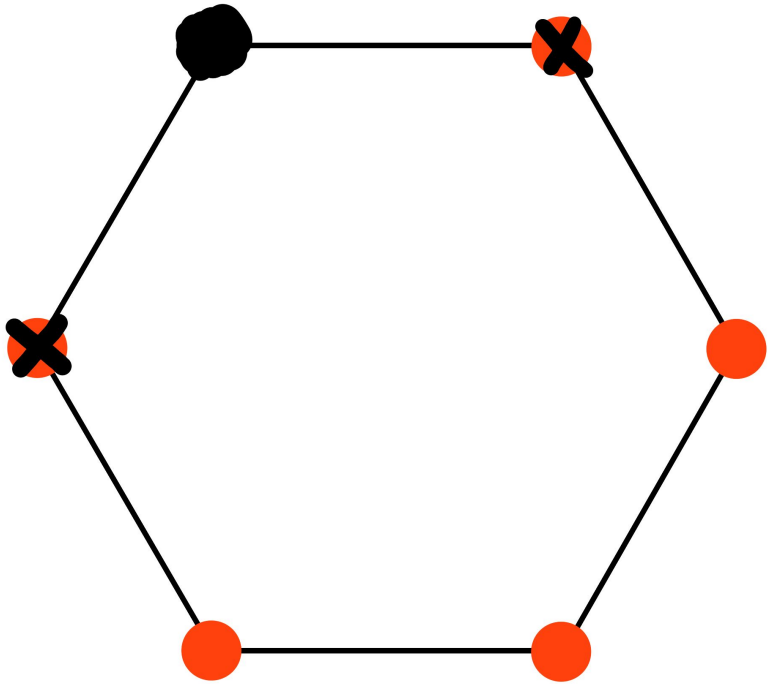
Maximizer's goal: Maximize the final number of colored vertices.



The Independence Coloring Game

Minimizer's goal: Minimize the final number of colored vertices.

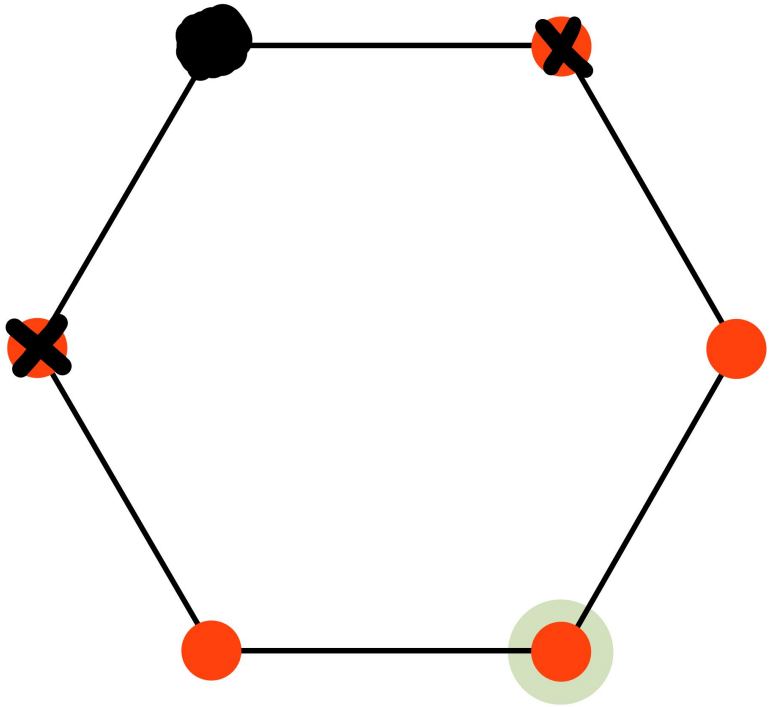
Maximizer's goal: Maximize the final number of colored vertices.



The Independence Coloring Game

Minimizer's goal: Minimize the final number of colored vertices.

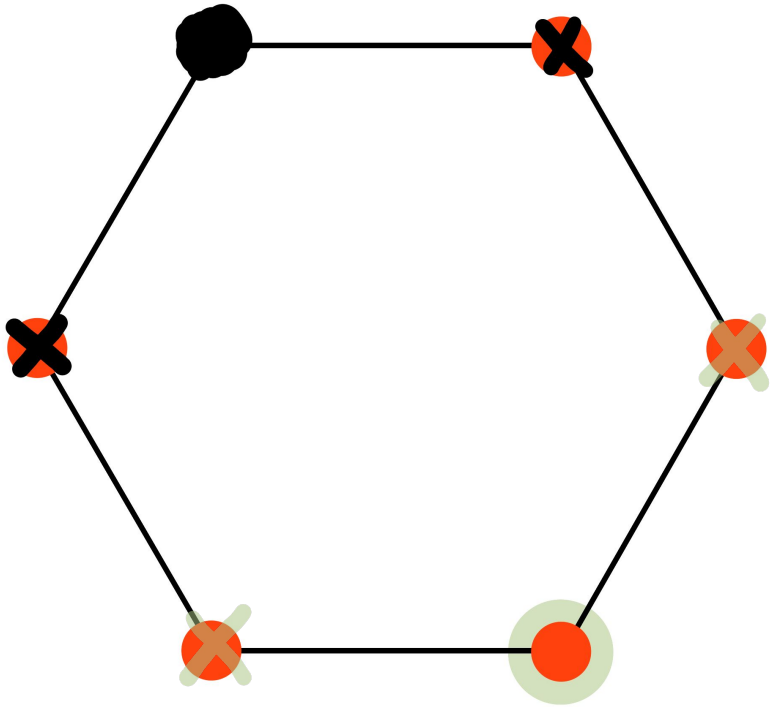
Maximizer's goal: Maximize the final number of colored vertices.



The Independence Coloring Game

Minimizer's goal: Minimize the final number of colored vertices.

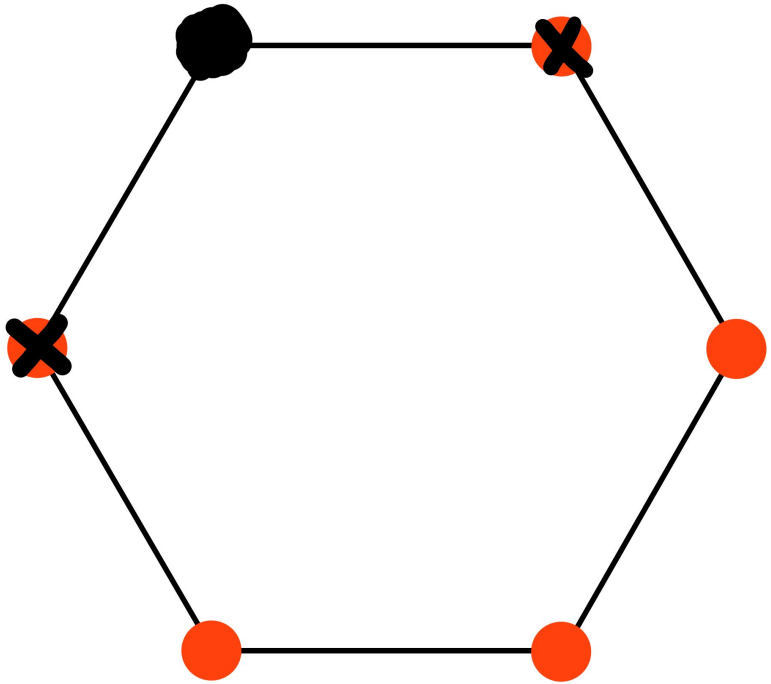
Maximizer's goal: Maximize the final number of colored vertices.



The Independence Coloring Game

Minimizer's goal: Minimize the final number of colored vertices.

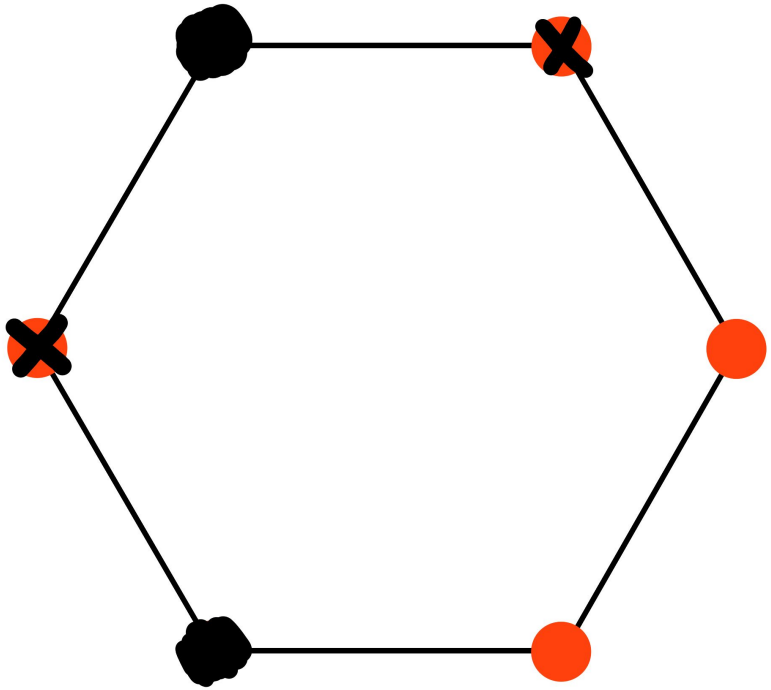
Maximizer's goal: Maximize the final number of colored vertices.



The Independence Coloring Game

Minimizer's goal: Minimize the final number of colored vertices.

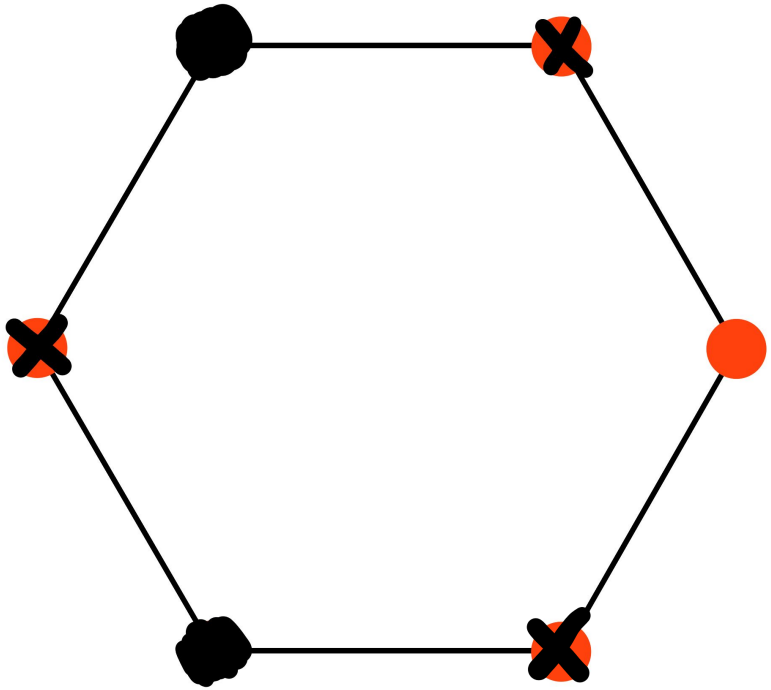
Maximizer's goal: Maximize the final number of colored vertices.



The Independence Coloring Game

Minimizer's goal: Minimize the final number of colored vertices.

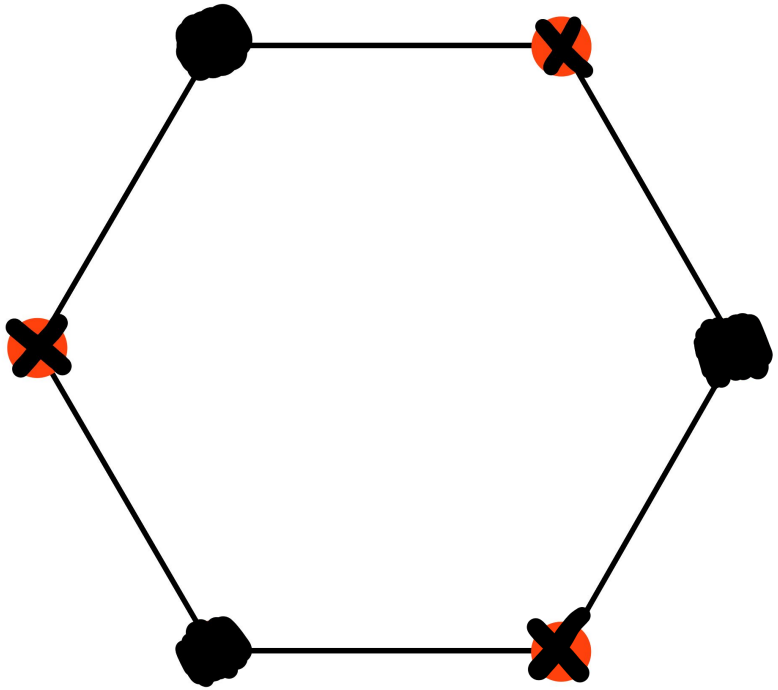
Maximizer's goal: Maximize the final number of colored vertices.



The Independence Coloring Game

Minimizer's goal: Minimize the final number of colored vertices.

Maximizer's goal: Maximize the final number of colored vertices.



How can we mathematically determine the most optimal move for both Minimizer and Maximizer in the independence coloring game?

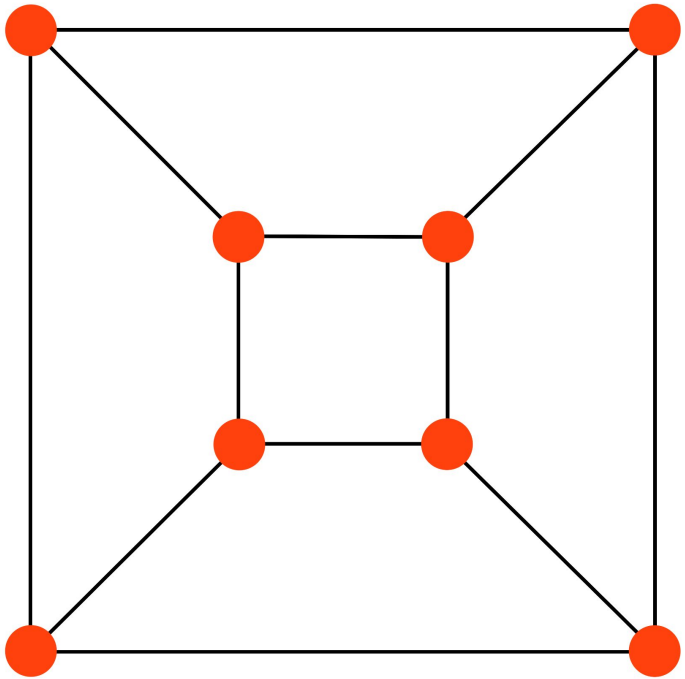
Minimizer’s Optimal Game-Play Strategy for Vertex-Transitive Graphs

Minimizer’s Optimal Game-Play Strategy for Vertex-Transitive Graphs

U(v) value: Represents the number of adjacent, unavailable (colored or adjacent to a colored vertex) vertices for an uncolored vertex.

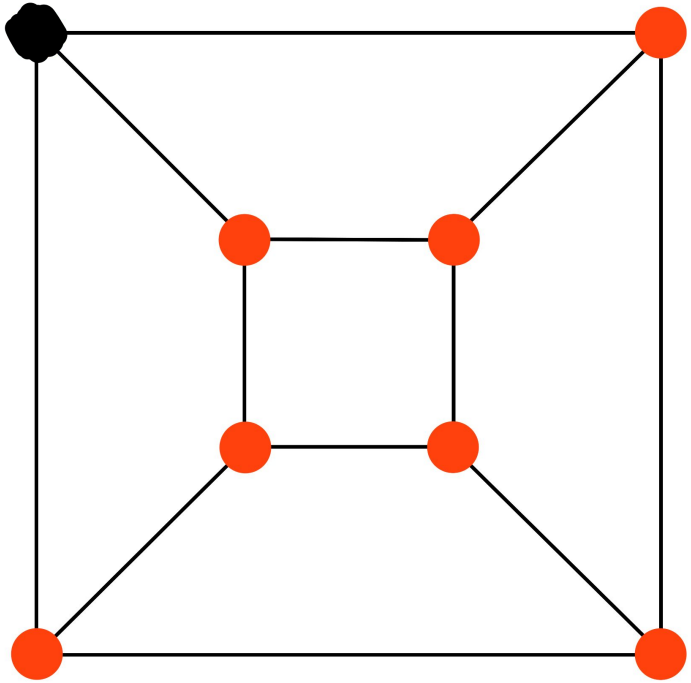
Minimizer's Optimal Game-Play Strategy for Vertex-Transitive Graphs

U(v) value: Represents the number of adjacent, unavailable (colored or adjacent to a colored vertex) vertices for an uncolored vertex.



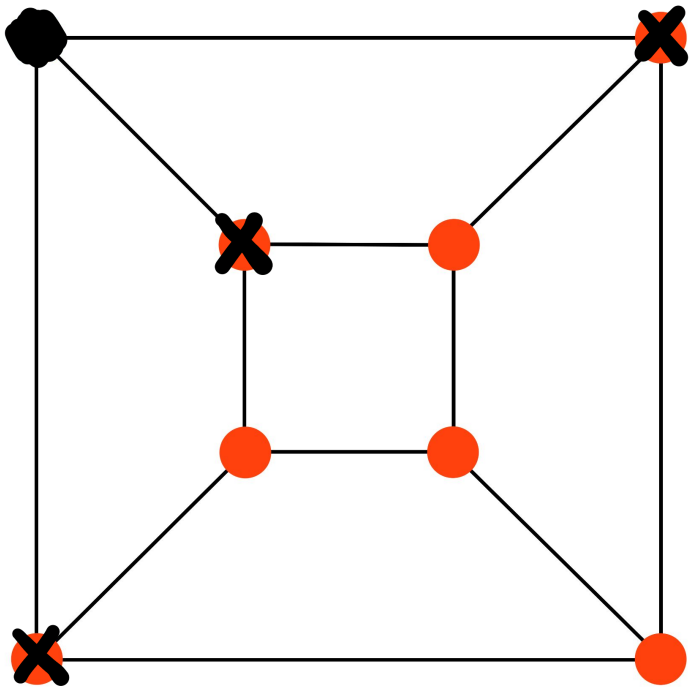
Minimizer's Optimal Game-Play Strategy for Vertex-Transitive Graphs

U(v) value: Represents the number of adjacent, unavailable (colored or adjacent to a colored vertex) vertices for an uncolored vertex.



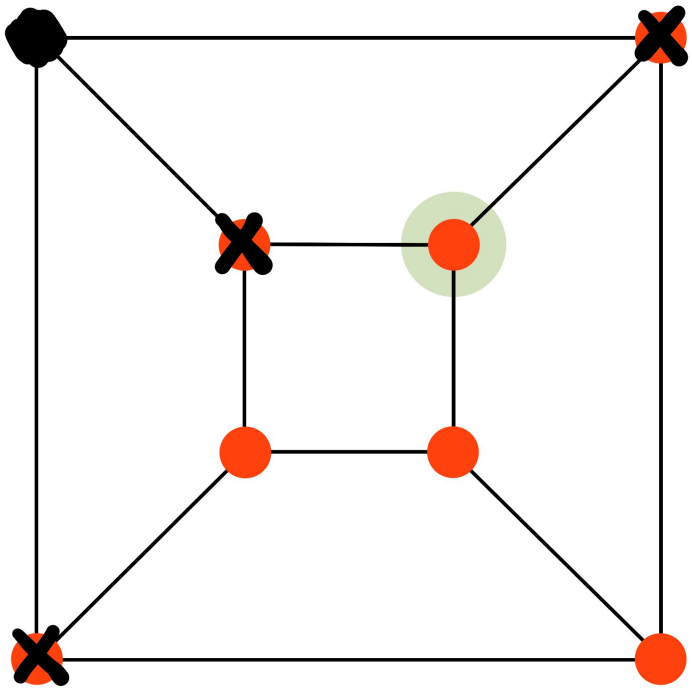
Minimizer's Optimal Game-Play Strategy for Vertex-Transitive Graphs

U(v) value: Represents the number of adjacent, unavailable (colored or adjacent to a colored vertex) vertices for an uncolored vertex.



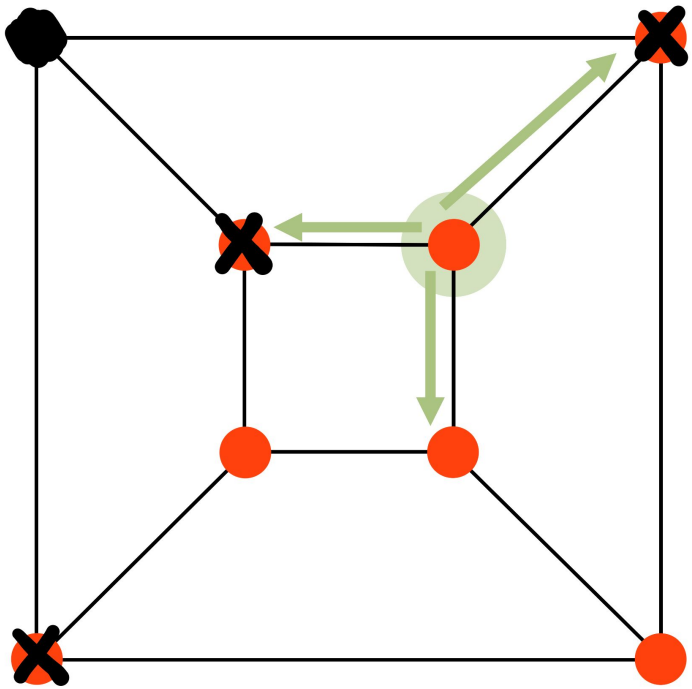
Minimizer’s Optimal Game-Play Strategy for Vertex-Transitive Graphs

U(v) value: Represents the number of adjacent, unavailable (colored or adjacent to a colored vertex) vertices for an uncolored vertex.



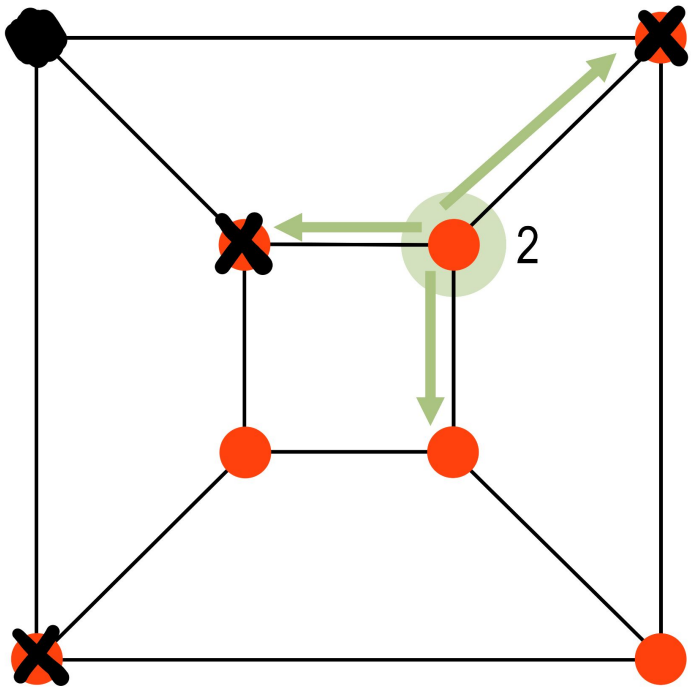
Minimizer's Optimal Game-Play Strategy for Vertex-Transitive Graphs

U(v) value: Represents the number of adjacent, unavailable (colored or adjacent to a colored vertex) vertices for an uncolored vertex.



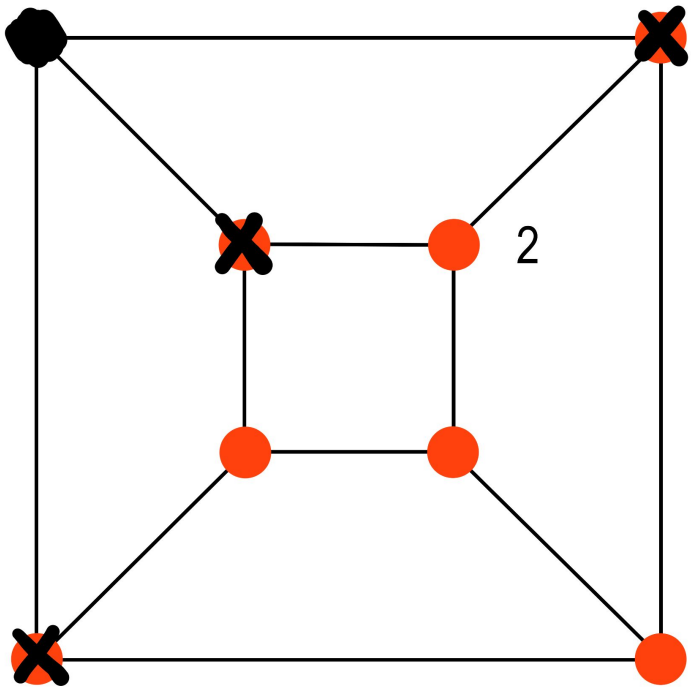
Minimizer's Optimal Game-Play Strategy for Vertex-Transitive Graphs

U(v) value: Represents the number of adjacent, unavailable (colored or adjacent to a colored vertex) vertices for an uncolored vertex.



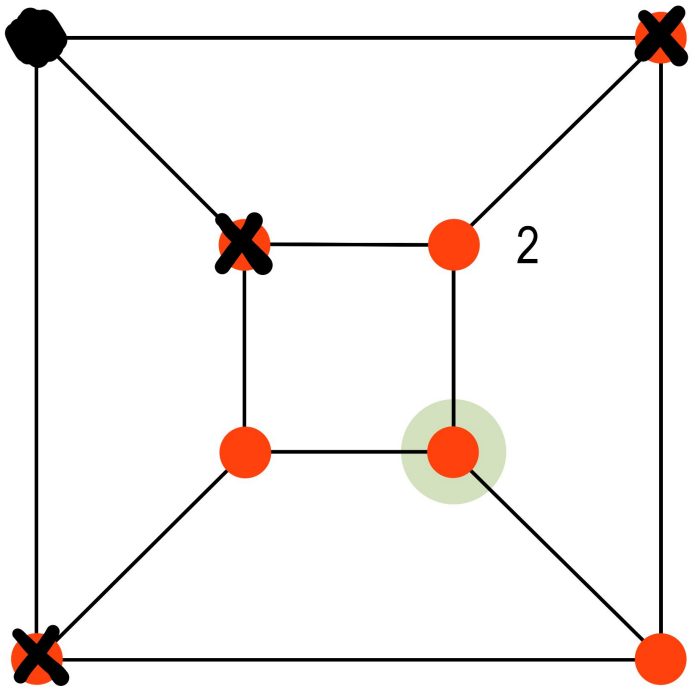
Minimizer's Optimal Game-Play Strategy for Vertex-Transitive Graphs

U(v) value: Represents the number of adjacent, unavailable (colored or adjacent to a colored vertex) vertices for an uncolored vertex.



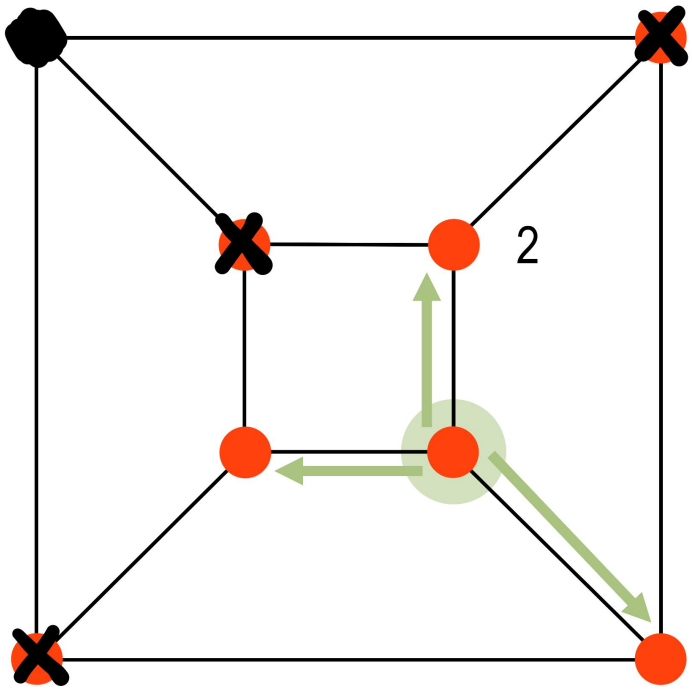
Minimizer's Optimal Game-Play Strategy for Vertex-Transitive Graphs

U(v) value: Represents the number of adjacent, unavailable (colored or adjacent to a colored vertex) vertices for an uncolored vertex.



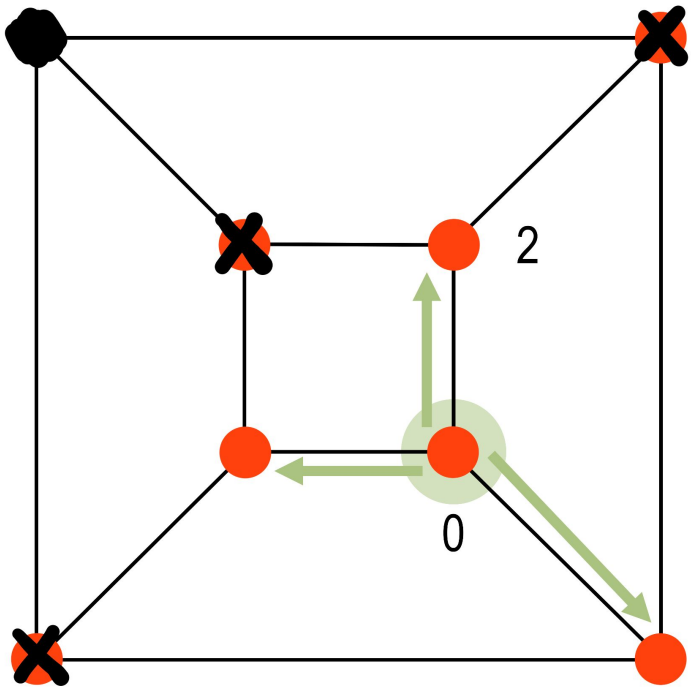
Minimizer's Optimal Game-Play Strategy for Vertex-Transitive Graphs

U(v) value: Represents the number of adjacent, unavailable (colored or adjacent to a colored vertex) vertices for an uncolored vertex.



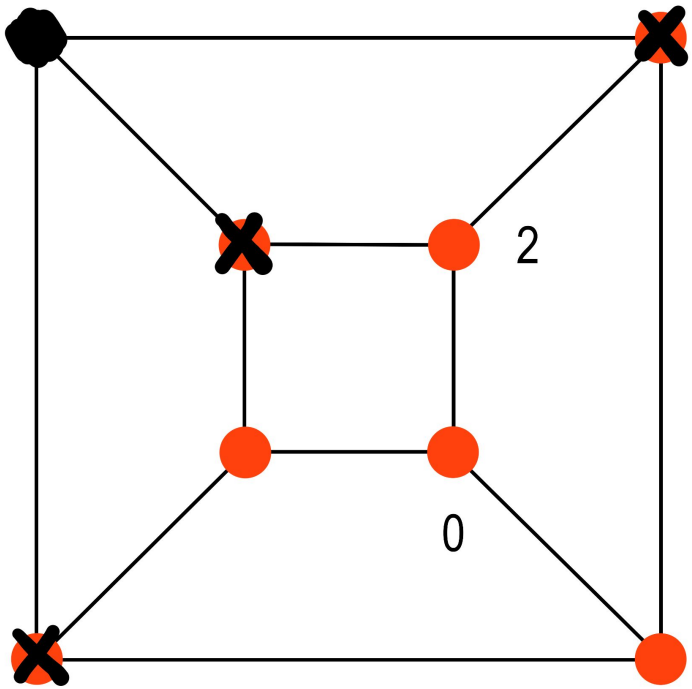
Minimizer's Optimal Game-Play Strategy for Vertex-Transitive Graphs

U(v) value: Represents the number of adjacent, unavailable (colored or adjacent to a colored vertex) vertices for an uncolored vertex.



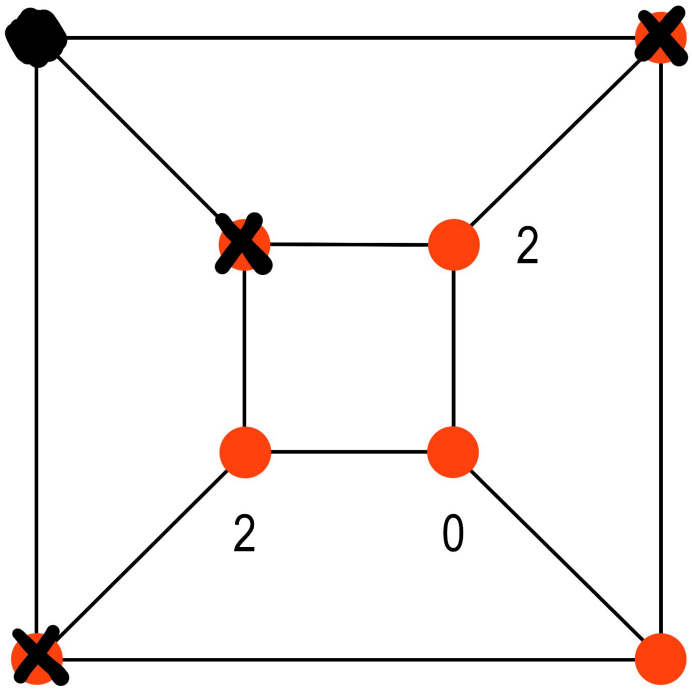
Minimizer's Optimal Game-Play Strategy for Vertex-Transitive Graphs

U(v) value: Represents the number of adjacent, unavailable (colored or adjacent to a colored vertex) vertices for an uncolored vertex.



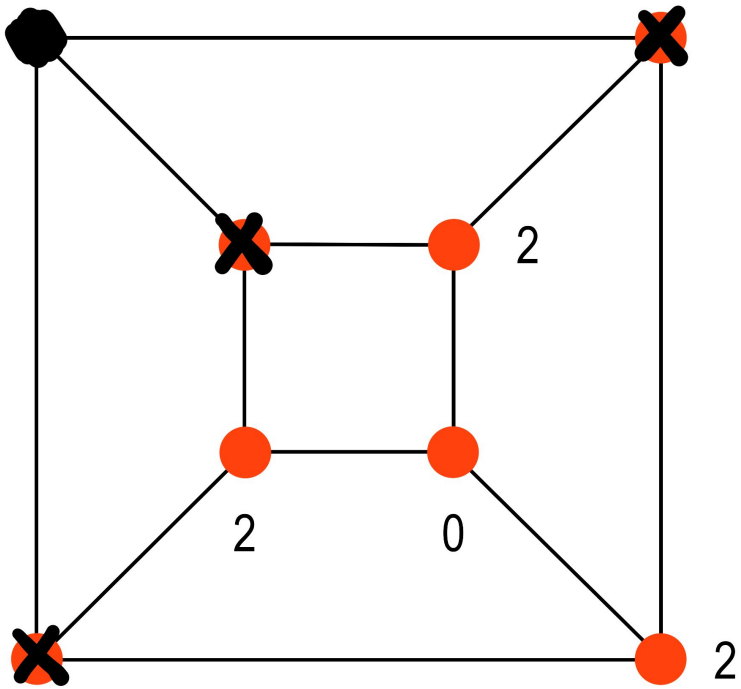
Minimizer's Optimal Game-Play Strategy for Vertex-Transitive Graphs

U(v) value: Represents the number of adjacent, unavailable (colored or adjacent to a colored vertex) vertices for an uncolored vertex.



Minimizer's Optimal Game-Play Strategy for Vertex-Transitive Graphs

U(v) value: Represents the number of adjacent, unavailable (colored or adjacent to a colored vertex) vertices for an uncolored vertex.

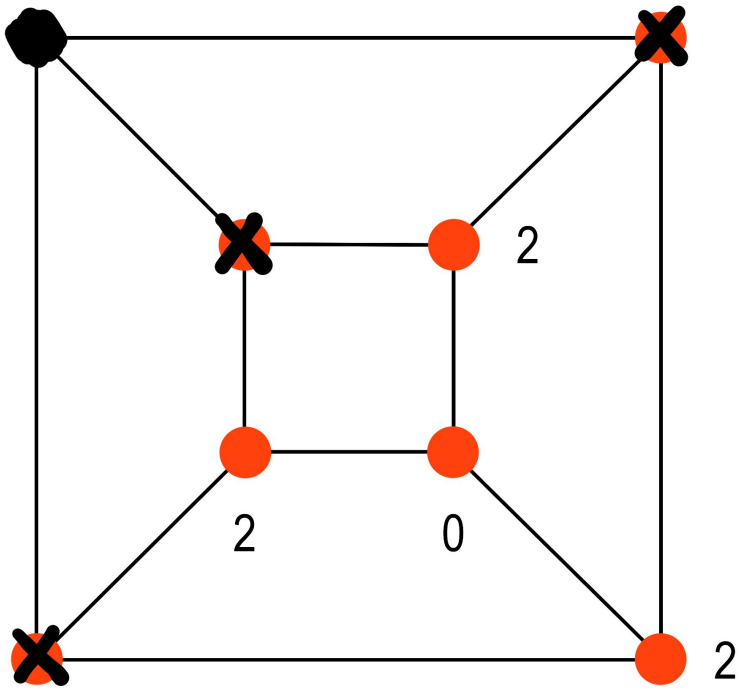


Minimizer's Optimal Game-Play Strategy for Vertex-Transitive Graphs

U(v) value: Represents the number of adjacent, unavailable (colored or adjacent to a colored vertex) vertices for an uncolored vertex.

<u>S(u) value</u> (underlined): Represents the sum of adjacent U(v) values for an uncolored vertex.

1. The <u>S(u) value</u> of a vertex includes the U(v) value of the vertex itself in addition to the U(v) values of its adjacent vertices.
2. The U(v) value for unavailable vertex is 0.

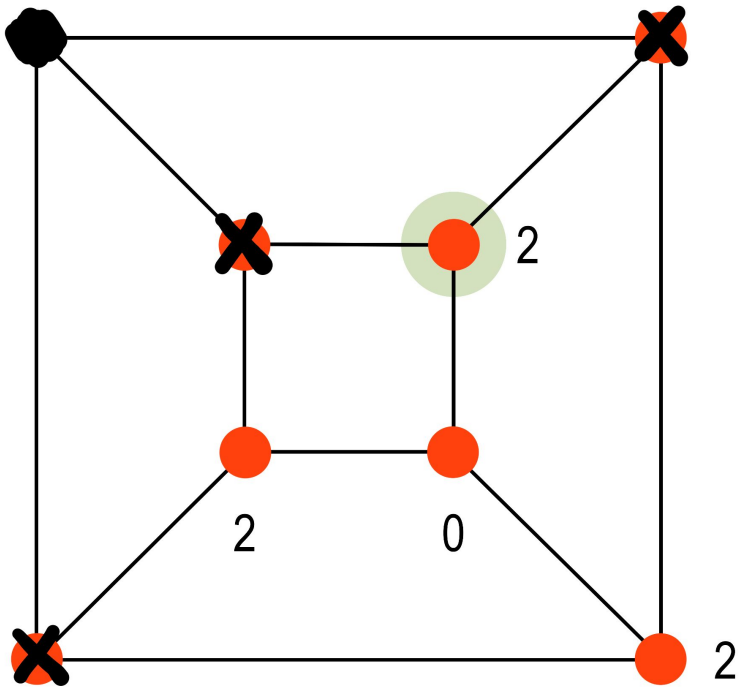


Minimizer's Optimal Game-Play Strategy for Vertex-Transitive Graphs

U(v) value: Represents the number of adjacent, unavailable (colored or adjacent to a colored vertex) vertices for an uncolored vertex.

<u>S(u) value</u> (underlined): Represents the sum of adjacent U(v) values for an uncolored vertex.

1. The <u>S(u) value</u> of a vertex includes the U(v) value of the vertex itself in addition to the U(v) values of its adjacent vertices.
2. The U(v) value for unavailable vertex is 0.

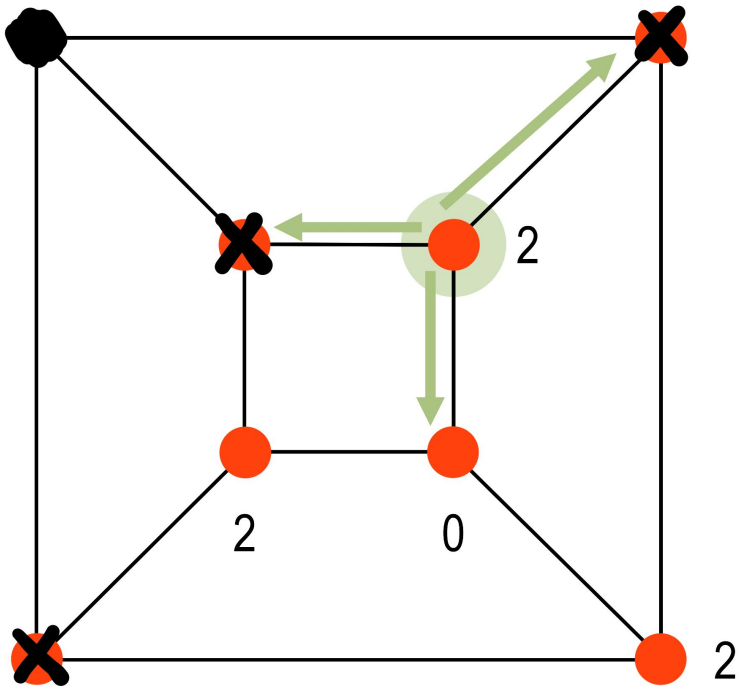


Minimizer's Optimal Game-Play Strategy for Vertex-Transitive Graphs

U(v) value: Represents the number of adjacent, unavailable (colored or adjacent to a colored vertex) vertices for an uncolored vertex.

<u>S(u) value</u> (underlined): Represents the sum of adjacent U(v) values for an uncolored vertex.

1. The <u>S(u) value</u> of a vertex includes the U(v) value of the vertex itself in addition to the U(v) values of its adjacent vertices.
2. The U(v) value for unavailable vertex is 0.

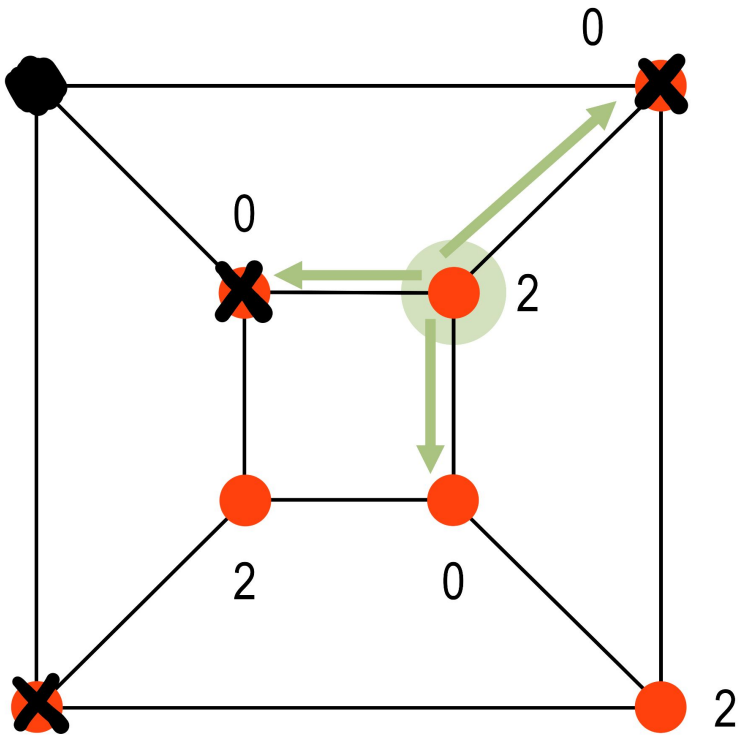


Minimizer's Optimal Game-Play Strategy for Vertex-Transitive Graphs

U(v) value: Represents the number of adjacent, unavailable (colored or adjacent to a colored vertex) vertices for an uncolored vertex.

<u>S(u) value</u> (underlined): Represents the sum of adjacent U(v) values for an uncolored vertex.

1. The <u>S(u) value</u> of a vertex includes the U(v) value of the vertex itself in addition to the U(v) values of its adjacent vertices.
2. The U(v) value for unavailable vertex is 0.

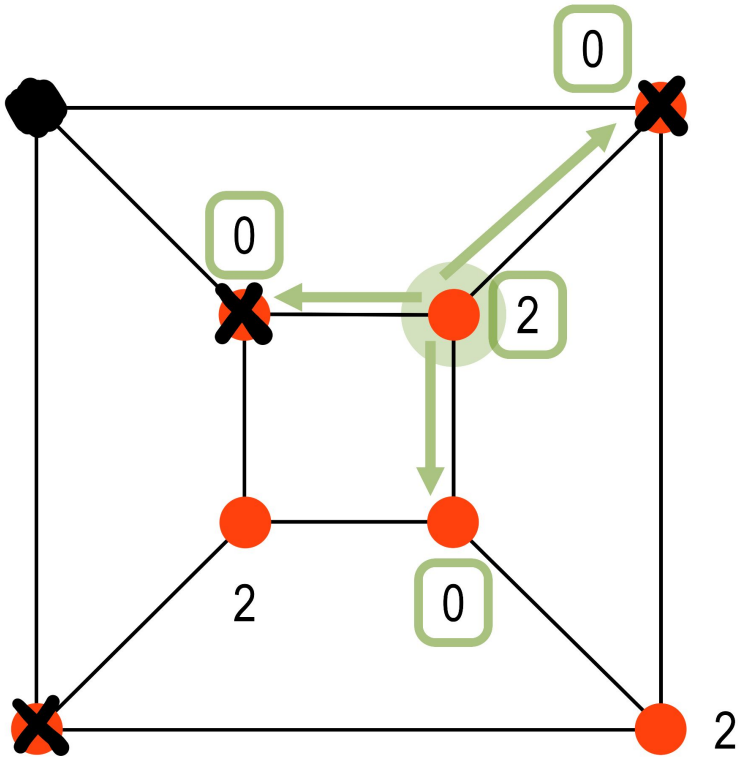


Minimizer's Optimal Game-Play Strategy for Vertex-Transitive Graphs

U(v) value: Represents the number of adjacent, unavailable (colored or adjacent to a colored vertex) vertices for an uncolored vertex.

<u>S(u) value</u> (underlined): Represents the sum of adjacent U(v) values for an uncolored vertex.

1. The <u>S(u) value</u> of a vertex includes the U(v) value of the vertex itself in addition to the U(v) values of its adjacent vertices.
2. The U(v) value for unavailable vertex is 0.

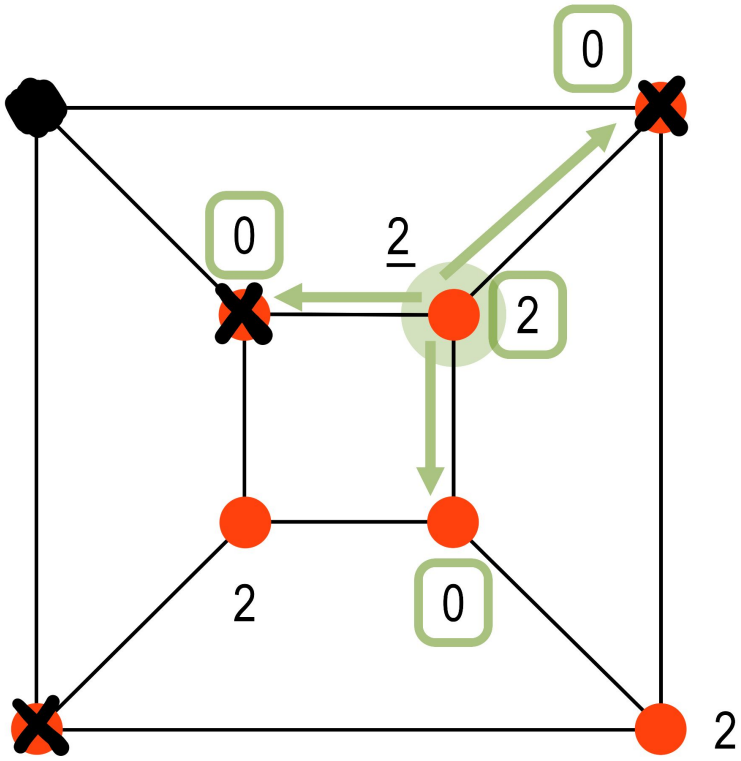


Minimizer's Optimal Game-Play Strategy for Vertex-Transitive Graphs

U(v) value: Represents the number of adjacent, unavailable (colored or adjacent to a colored vertex) vertices for an uncolored vertex.

<u>S(u) value</u> (underlined): Represents the sum of adjacent U(v) values for an uncolored vertex.

1. The <u>S(u) value</u> of a vertex includes the U(v) value of the vertex itself in addition to the U(v) values of its adjacent vertices.
2. The U(v) value for unavailable vertex is 0.

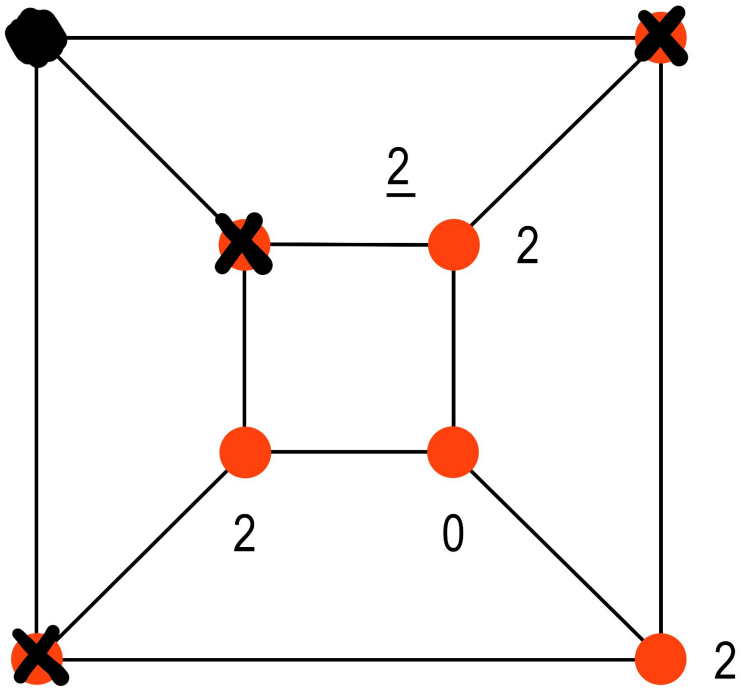


Minimizer's Optimal Game-Play Strategy for Vertex-Transitive Graphs

U(v) value: Represents the number of adjacent, unavailable (colored or adjacent to a colored vertex) vertices for an uncolored vertex.

<u>S(u) value</u> (underlined): Represents the sum of adjacent U(v) values for an uncolored vertex.

1. The <u>S(u) value</u> of a vertex includes the U(v) value of the vertex itself in addition to the U(v) values of its adjacent vertices.
2. The U(v) value for unavailable vertex is 0.

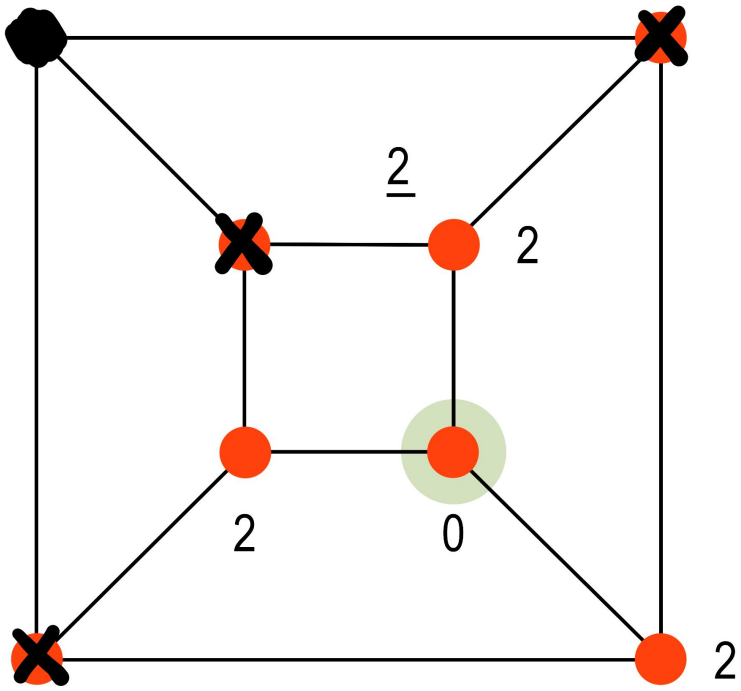


Minimizer's Optimal Game-Play Strategy for Vertex-Transitive Graphs

U(v) value: Represents the number of adjacent, unavailable (colored or adjacent to a colored vertex) vertices for an uncolored vertex.

<u>S(u) value</u> (underlined): Represents the sum of adjacent U(v) values for an uncolored vertex.

1. The <u>S(u) value</u> of a vertex includes the U(v) value of the vertex itself in addition to the U(v) values of its adjacent vertices.
2. The U(v) value for unavailable vertex is 0.

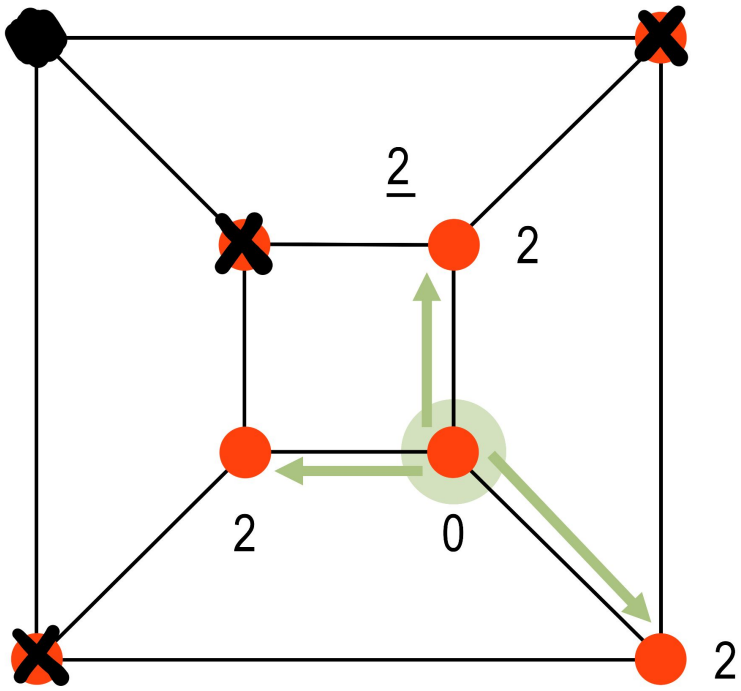


Minimizer's Optimal Game-Play Strategy for Vertex-Transitive Graphs

U(v) value: Represents the number of adjacent, unavailable (colored or adjacent to a colored vertex) vertices for an uncolored vertex.

<u>S(u) value</u> (underlined): Represents the sum of adjacent U(v) values for an uncolored vertex.

1. The <u>S(u) value</u> of a vertex includes the U(v) value of the vertex itself in addition to the U(v) values of its adjacent vertices.
2. The U(v) value for unavailable vertex is 0.

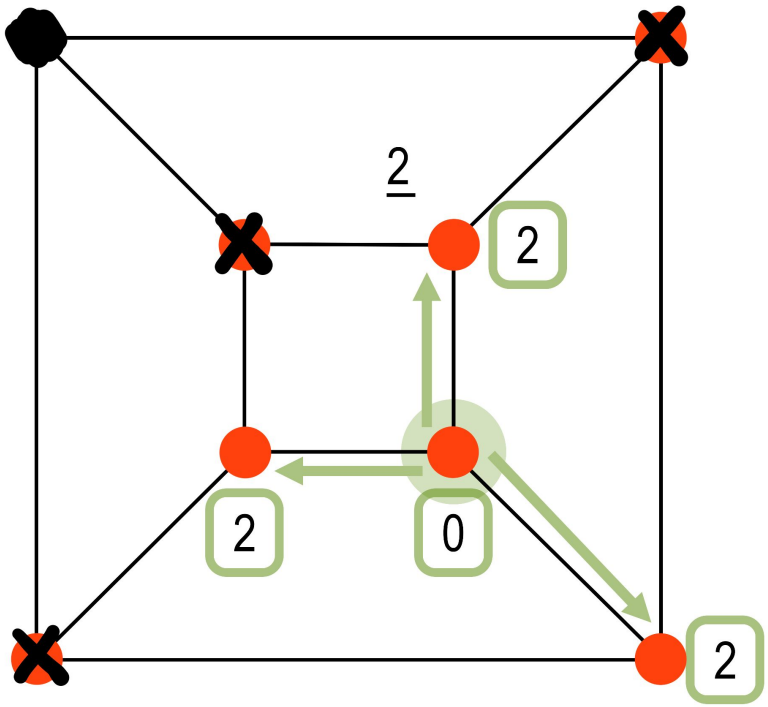


Minimizer's Optimal Game-Play Strategy for Vertex-Transitive Graphs

U(v) value: Represents the number of adjacent, unavailable (colored or adjacent to a colored vertex) vertices for an uncolored vertex.

<u>S(u) value</u> (underlined): Represents the sum of adjacent U(v) values for an uncolored vertex.

1. The <u>S(u) value</u> of a vertex includes the U(v) value of the vertex itself in addition to the U(v) values of its adjacent vertices.
2. The U(v) value for unavailable vertex is 0.

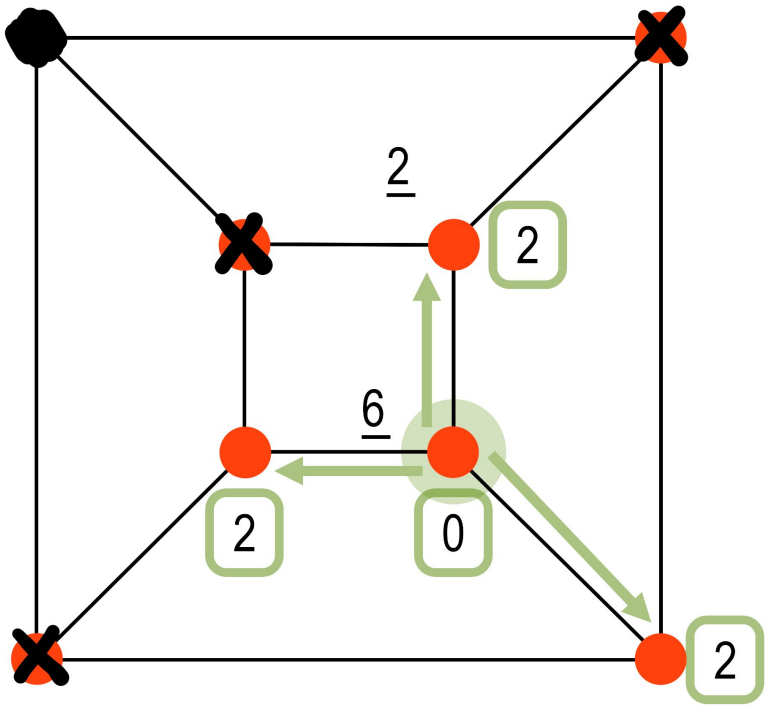


Minimizer's Optimal Game-Play Strategy for Vertex-Transitive Graphs

U(v) value: Represents the number of adjacent, unavailable (colored or adjacent to a colored vertex) vertices for an uncolored vertex.

<u>S(u) value</u> (underlined): Represents the sum of adjacent U(v) values for an uncolored vertex.

1. The <u>S(u) value</u> of a vertex includes the U(v) value of the vertex itself in addition to the U(v) values of its adjacent vertices.
2. The U(v) value for unavailable vertex is 0.

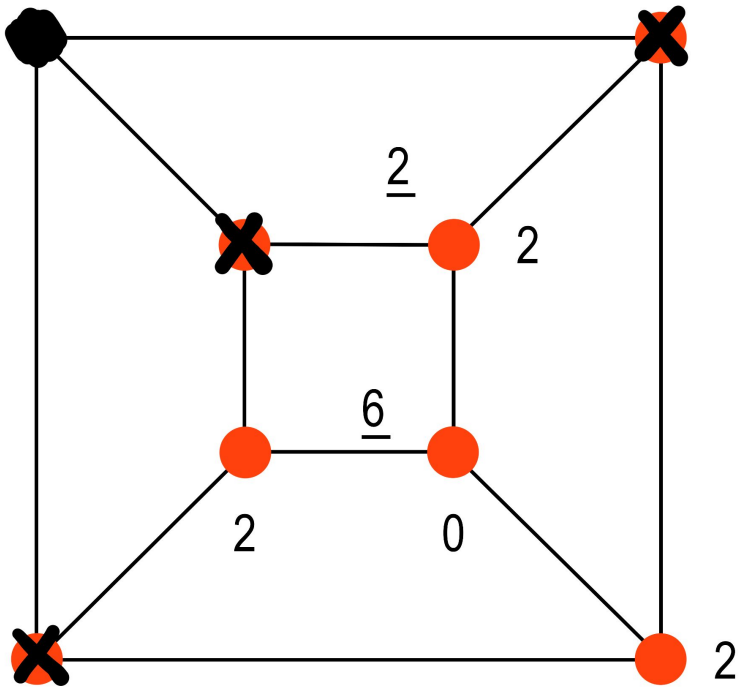


Minimizer's Optimal Game-Play Strategy for Vertex-Transitive Graphs

U(v) value: Represents the number of adjacent, unavailable (colored or adjacent to a colored vertex) vertices for an uncolored vertex.

<u>S(u) value</u> (underlined): Represents the sum of adjacent U(v) values for an uncolored vertex.

1. The <u>S(u) value</u> of a vertex includes the U(v) value of the vertex itself in addition to the U(v) values of its adjacent vertices.
2. The U(v) value for unavailable vertex is 0.

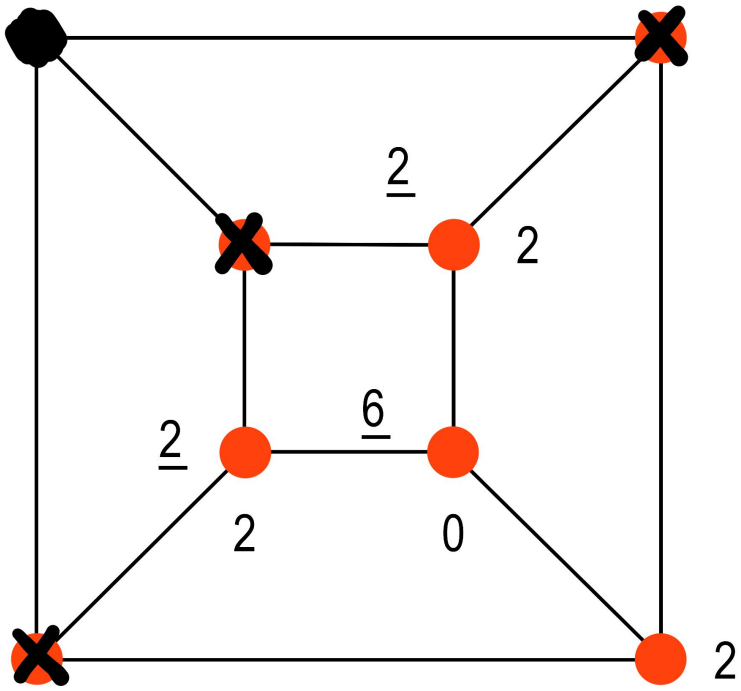


Minimizer's Optimal Game-Play Strategy for Vertex-Transitive Graphs

U(v) value: Represents the number of adjacent, unavailable (colored or adjacent to a colored vertex) vertices for an uncolored vertex.

<u>S(u) value</u> (underlined): Represents the sum of adjacent U(v) values for an uncolored vertex.

1. The <u>S(u) value</u> of a vertex includes the U(v) value of the vertex itself in addition to the U(v) values of its adjacent vertices.
2. The U(v) value for unavailable vertex is 0.

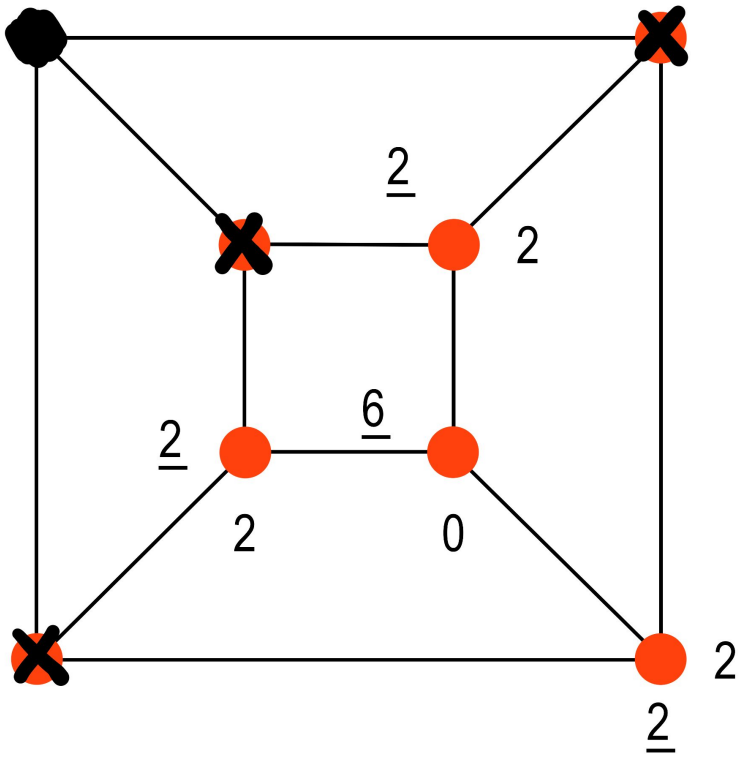


Minimizer's Optimal Game-Play Strategy for Vertex-Transitive Graphs

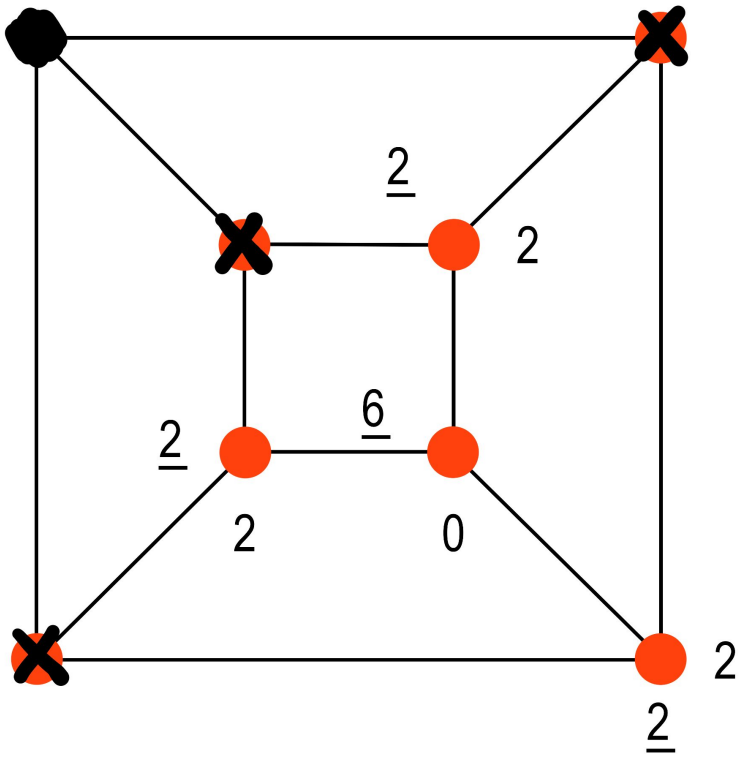
U(v) value: Represents the number of adjacent, unavailable (colored or adjacent to a colored vertex) vertices for an uncolored vertex.

<u>S(u) value</u> (underlined): Represents the sum of adjacent U(v) values for an uncolored vertex.

1. The <u>S(u) value</u> of a vertex includes the U(v) value of the vertex itself in addition to the U(v) values of its adjacent vertices.
2. The U(v) value for unavailable vertex is 0.

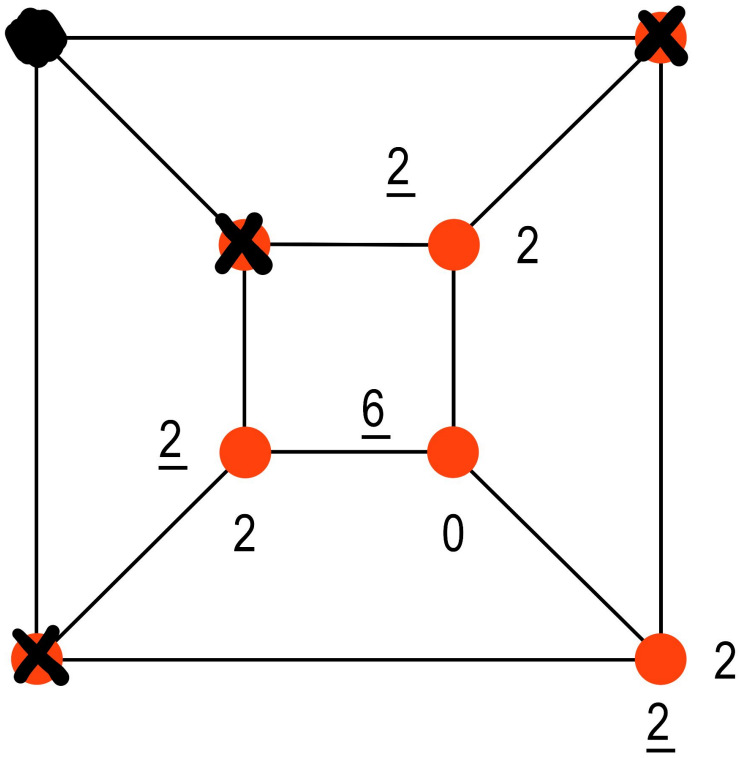


Minimizer's Optimal Game-Play Strategy for Vertex-Transitive Graphs



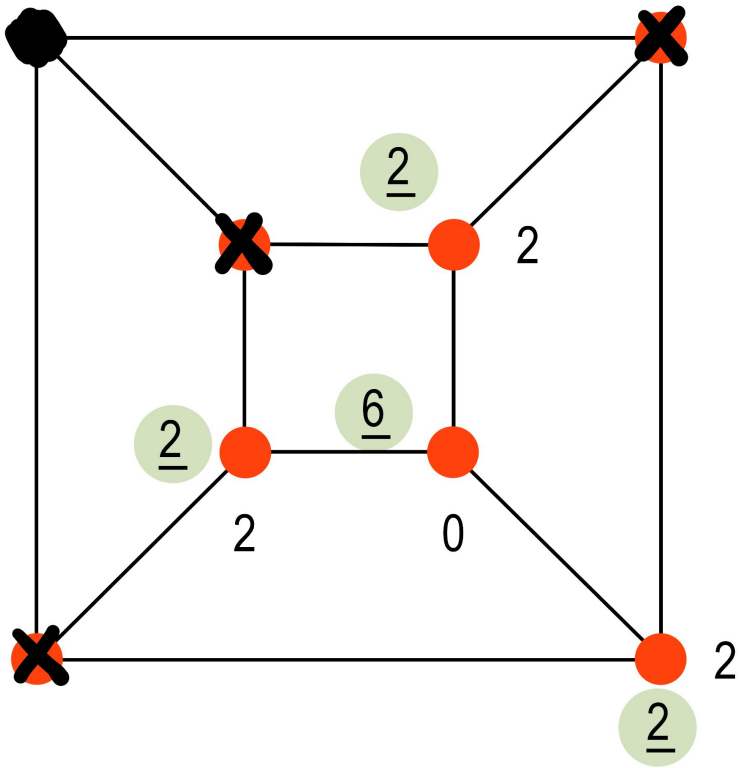
Minimizer's Optimal Game-Play Strategy for Vertex-Transitive Graphs

Step 1: Find all vertices with the **greatest** S(u) value.



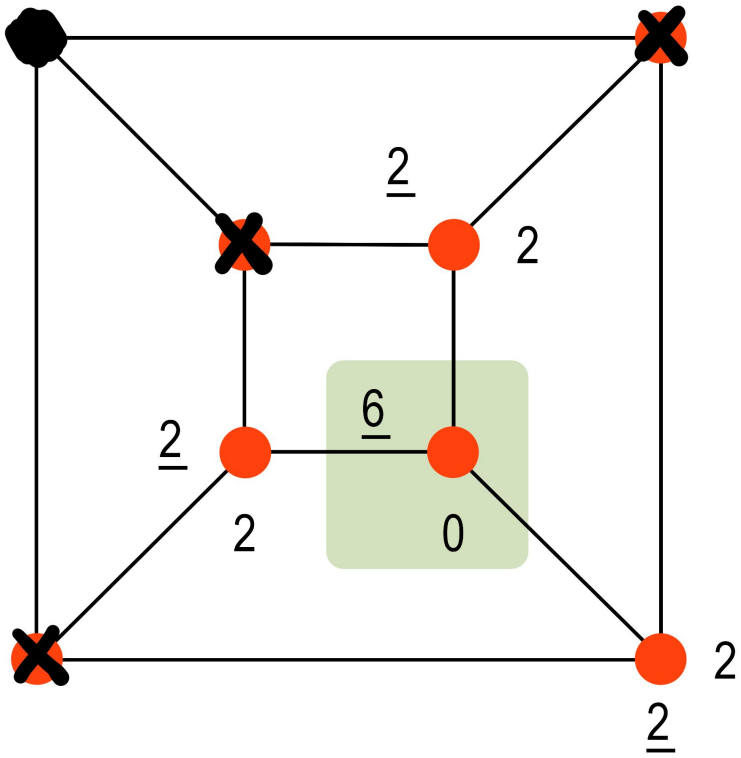
Minimizer's Optimal Game-Play Strategy for Vertex-Transitive Graphs

Step 1: Find all vertices with the **greatest** S(u) value.



Minimizer's Optimal Game-Play Strategy for Vertex-Transitive Graphs

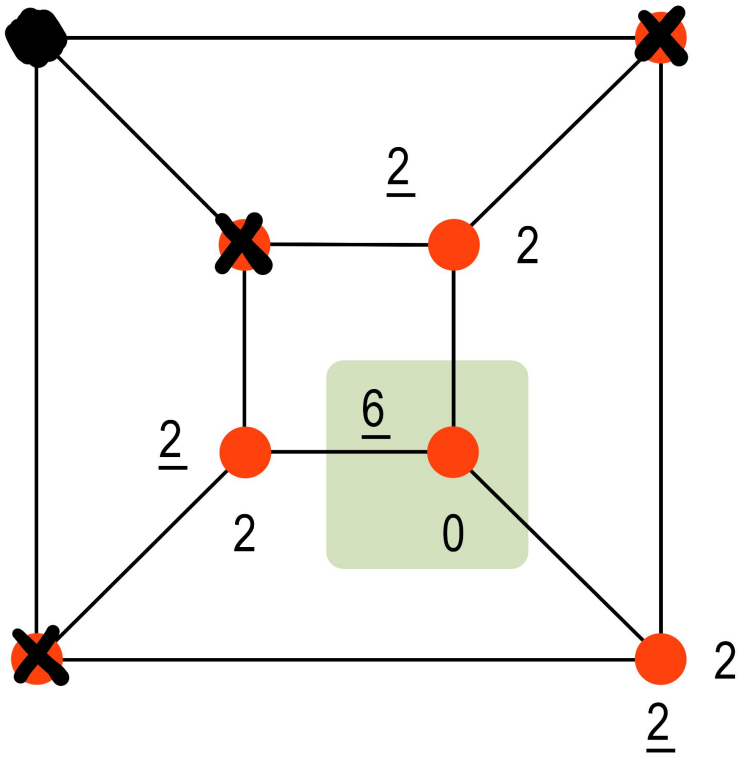
Step 1: Find all vertices with the **greatest** S(u) value.



Minimizer's Optimal Game-Play Strategy for Vertex-Transitive Graphs

Step 1: Find all vertices with the **greatest** S(u) value.

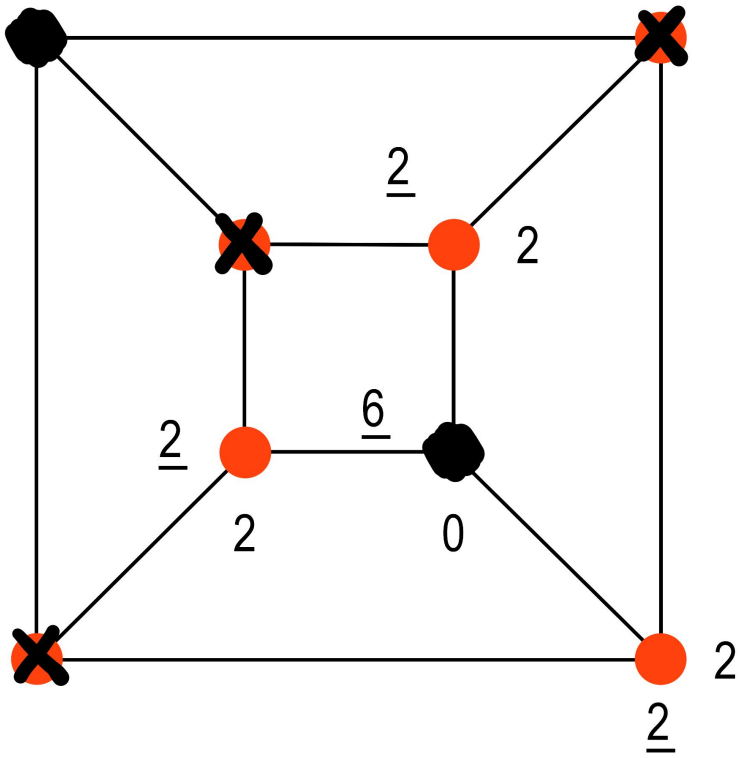
Step 2: From this subset, Minimizer then selects the vertex with the **lowest** U(v) value. If more than one vertex meets these criteria, Minimizer can randomly select any one of these vertices.



Minimizer's Optimal Game-Play Strategy for Vertex-Transitive Graphs

Step 1: Find all vertices with the **greatest** S(u) value.

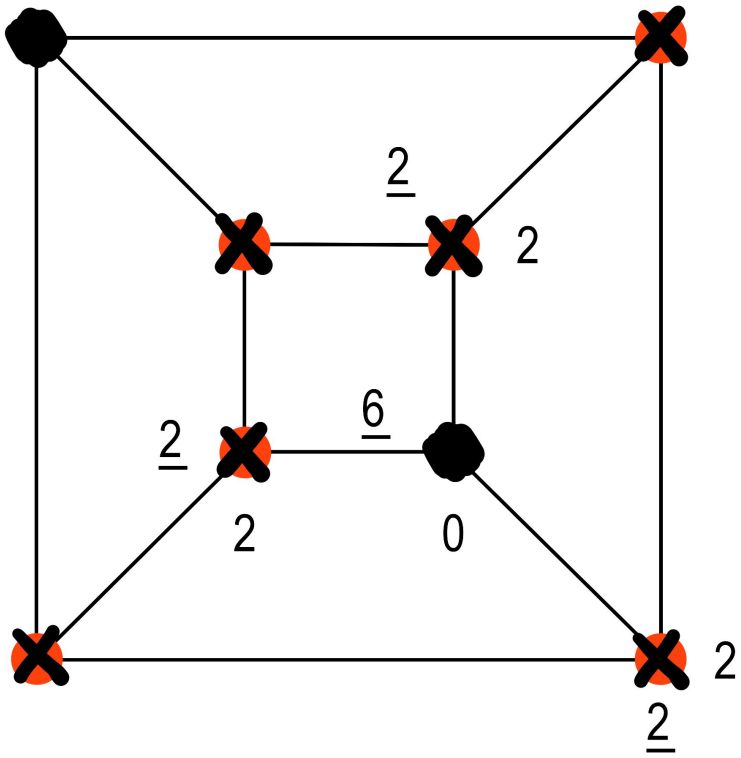
Step 2: From this subset, Minimizer then selects the vertex with the **lowest** U(v) value. If more than one vertex meets these criteria, Minimizer can randomly select any one of these vertices.



Minimizer's Optimal Game-Play Strategy for Vertex-Transitive Graphs

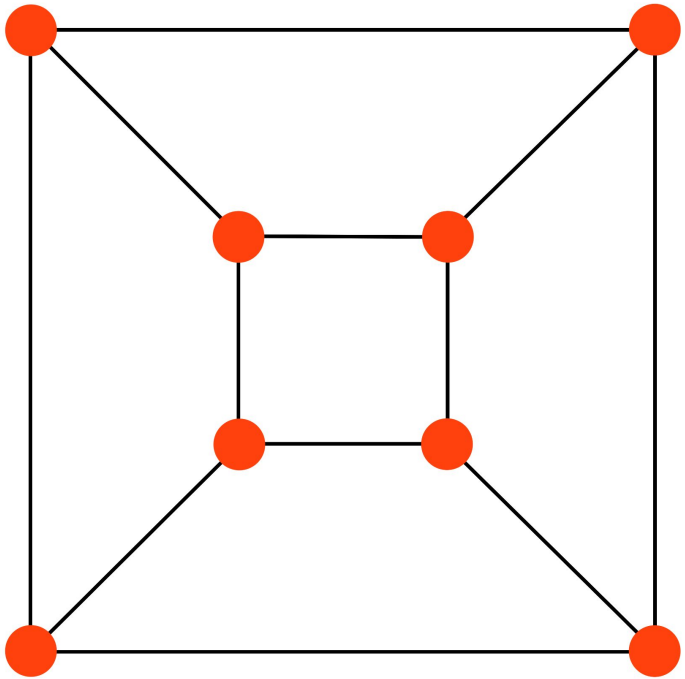
Step 1: Find all vertices with the **greatest** S(u) value.

Step 2: From this subset, Minimizer then selects the vertex with the **lowest** U(v) value. If more than one vertex meets these criteria, Minimizer can randomly select any one of these vertices.

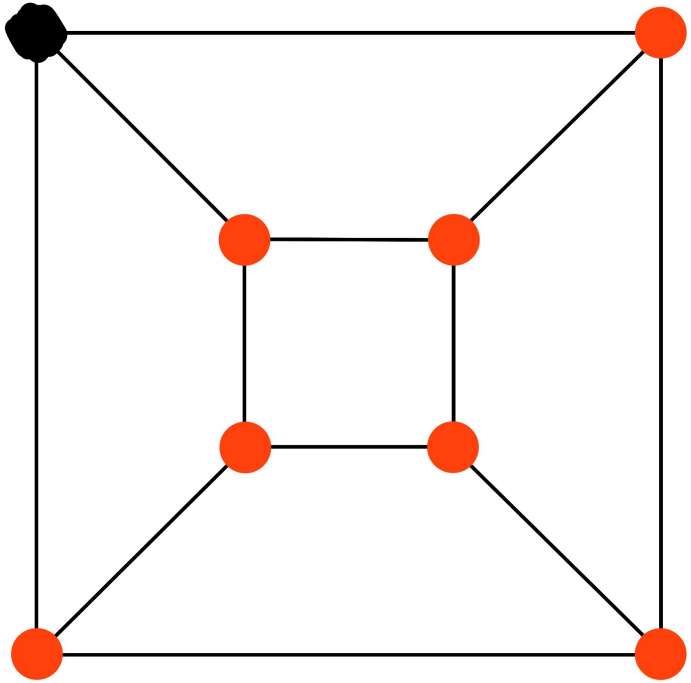


Maximizer’s Optimal Game-Play Strategy for Vertex Transitive Graphs

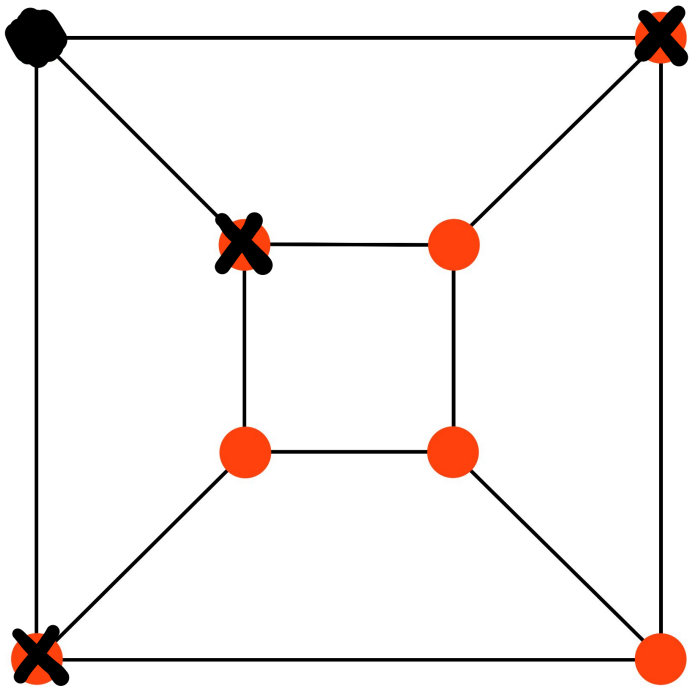
Maximizer's Optimal Game-Play Strategy for Vertex Transitive Graphs



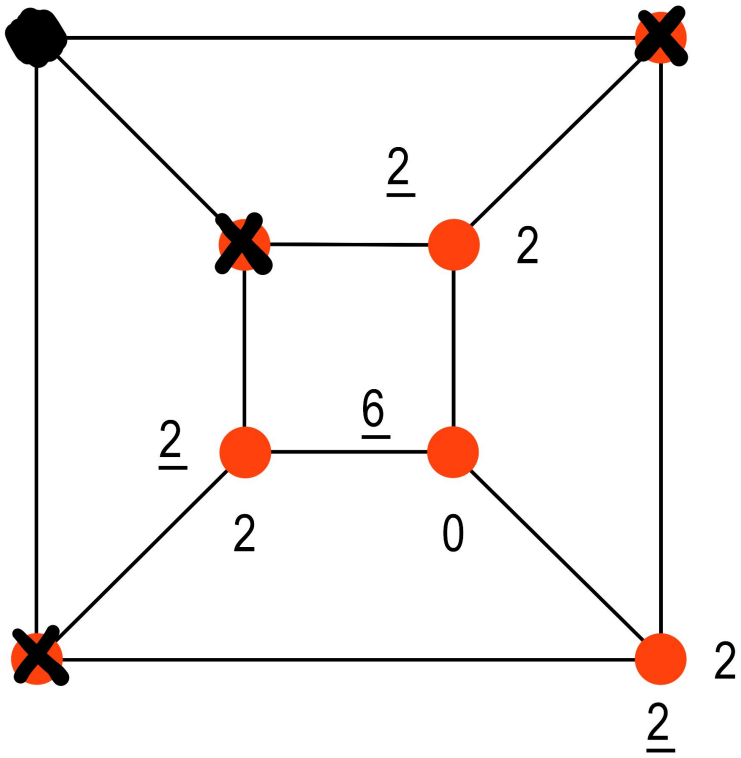
Maximizer's Optimal Game-Play Strategy for Vertex Transitive Graphs



Maximizer's Optimal Game-Play Strategy for Vertex Transitive Graphs

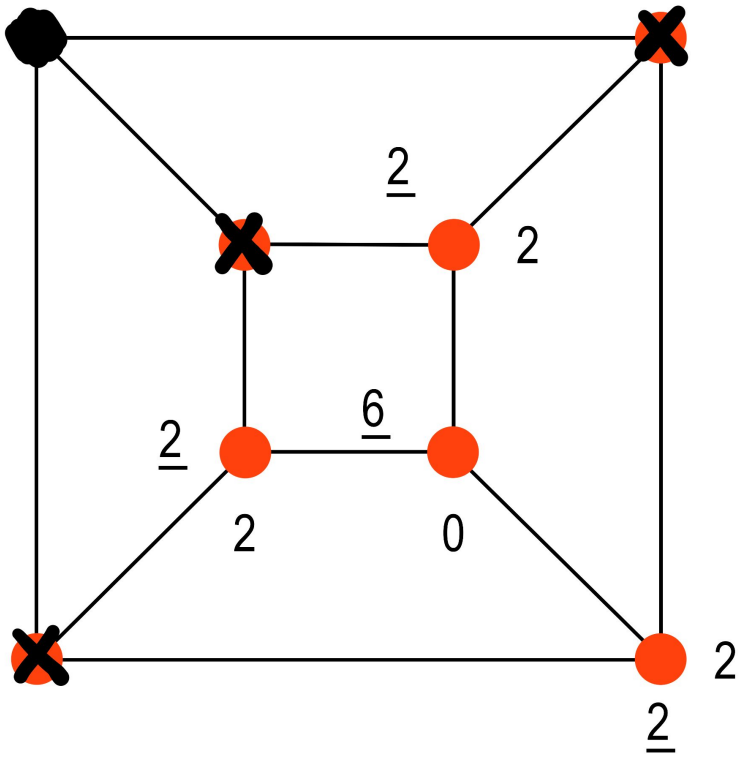


Maximizer's Optimal Game-Play Strategy for Vertex Transitive Graphs



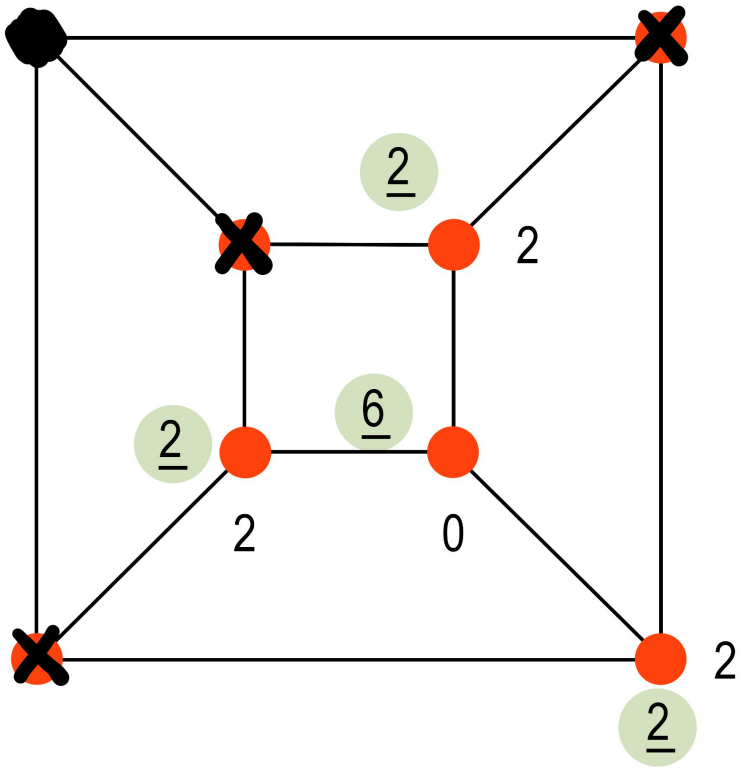
Maximizer's Optimal Game-Play Strategy for Vertex Transitive Graphs

Step 1: Find all vertices with the **lowest** S(u) value.



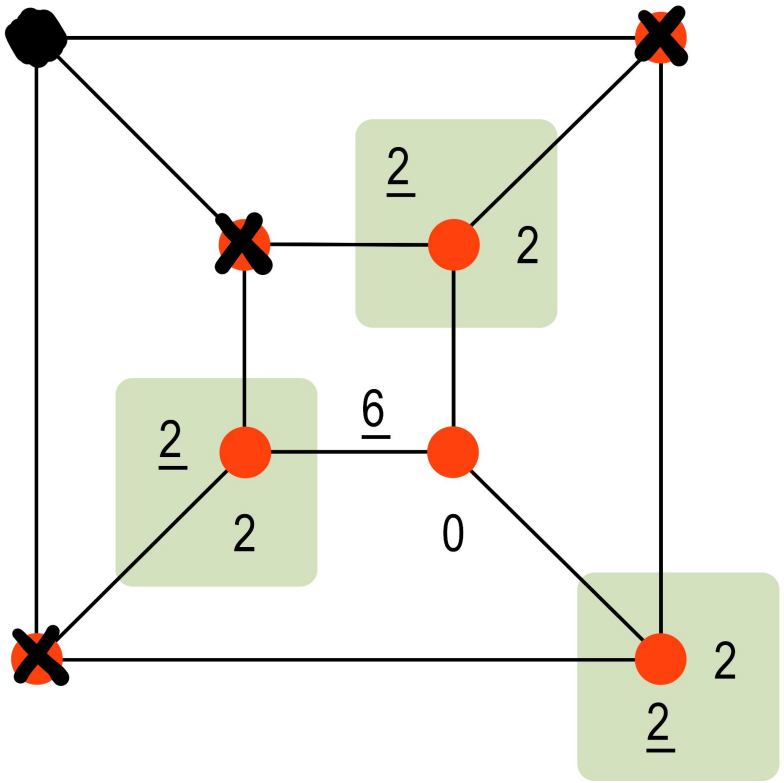
Maximizer's Optimal Game-Play Strategy for Vertex Transitive Graphs

Step 1: Find all vertices with the **lowest** S(u) value.



Maximizer's Optimal Game-Play Strategy for Vertex Transitive Graphs

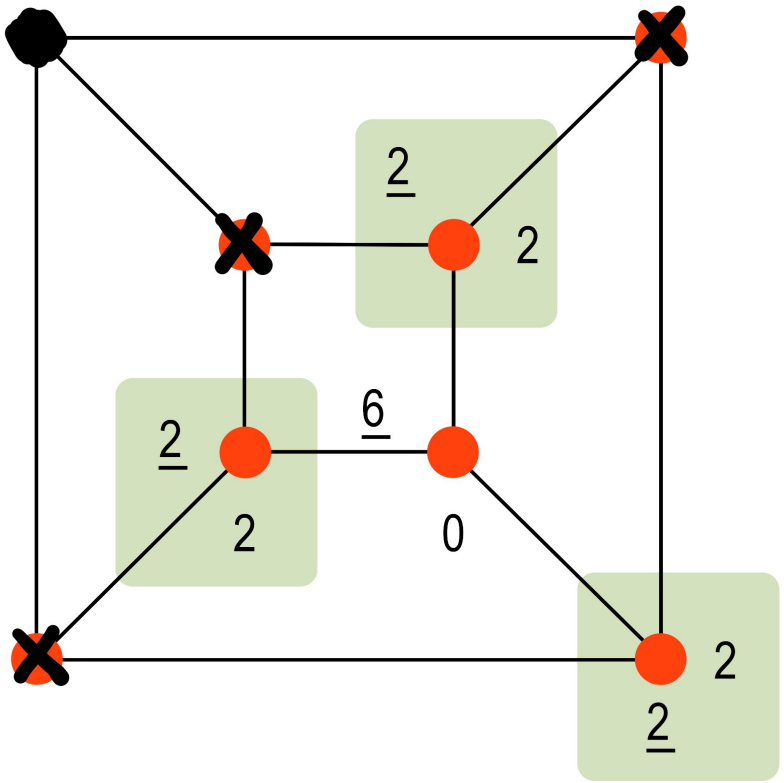
Step 1: Find all vertices with the **lowest** S(u) value.



Maximizer's Optimal Game-Play Strategy for Vertex Transitive Graphs

Step 1: Find all vertices with the **lowest** S(u) value.

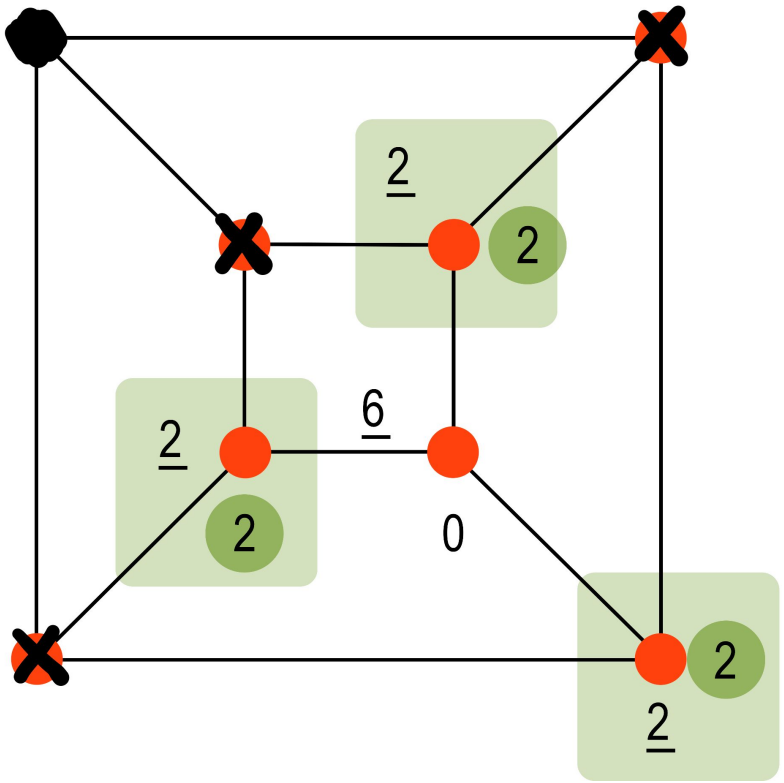
Step 2: From this subset, Maximizer then selects the vertex with the **greatest** U(v) value. If more than one vertex meets these criteria, Maximizer can randomly select any one of these vertices.



Maximizer's Optimal Game-Play Strategy for Vertex Transitive Graphs

Step 1: Find all vertices with the **lowest** S(u) value.

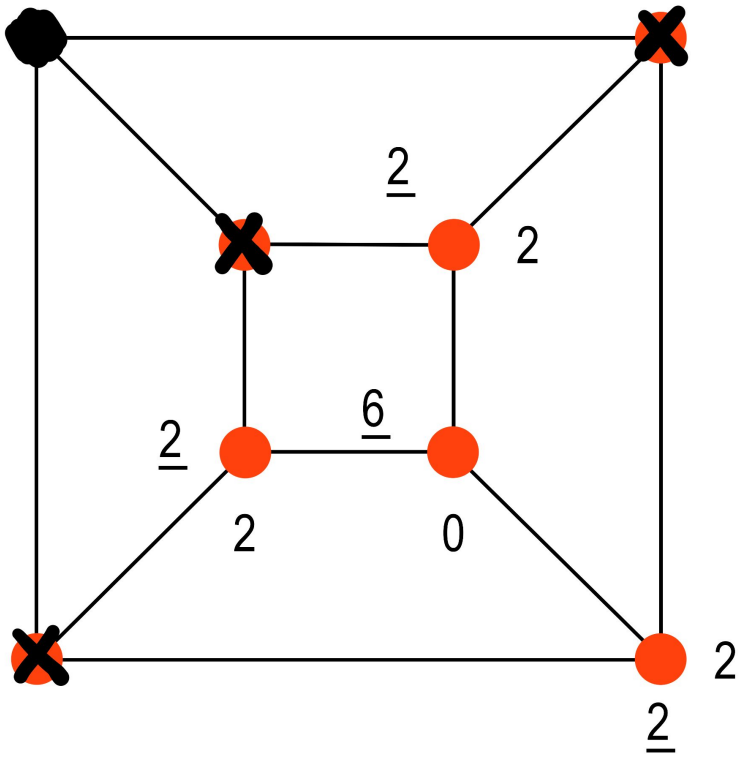
Step 2: From this subset, Maximizer then selects the vertex with the **greatest** U(v) value. If more than one vertex meets these criteria, Maximizer can randomly select any one of these vertices.



Maximizer's Optimal Game-Play Strategy for Vertex Transitive Graphs

Step 1: Find all vertices with the **lowest** S(u) value.

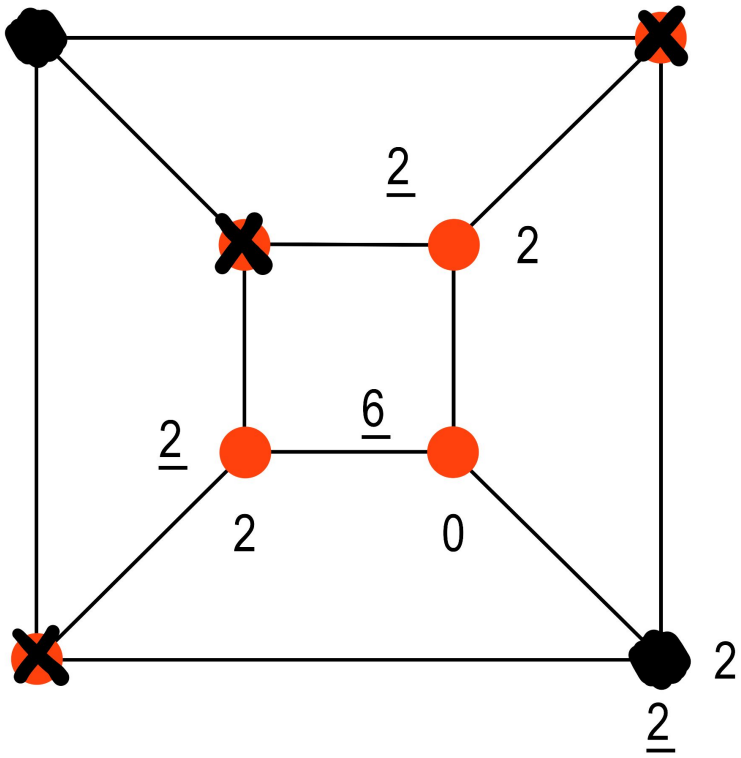
Step 2: From this subset, Maximizer then selects the vertex with the **greatest** U(v) value. If more than one vertex meets these criteria, Maximizer can randomly select any one of these vertices.



Maximizer's Optimal Game-Play Strategy for Vertex Transitive Graphs

Step 1: Find all vertices with the **lowest** S(u) value.

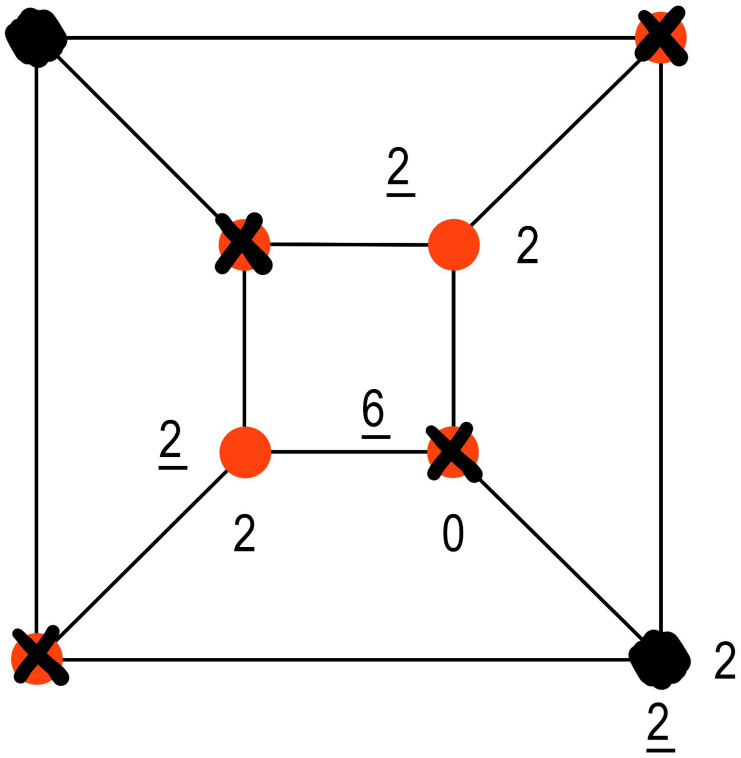
Step 2: From this subset, Maximizer then selects the vertex with the **greatest** U(v) value. If more than one vertex meets these criteria, Maximizer can randomly select any one of these vertices.



Maximizer's Optimal Game-Play Strategy for Vertex Transitive Graphs

Step 1: Find all vertices with the **lowest** S(u) value.

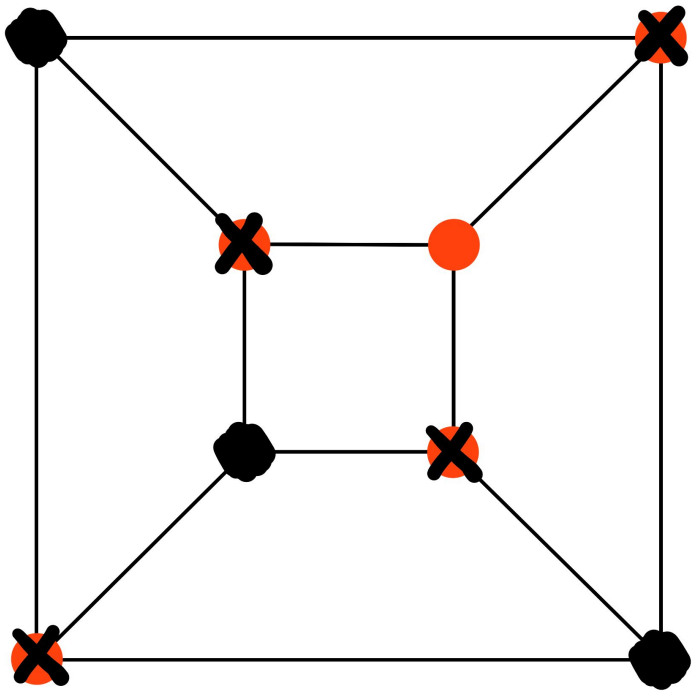
Step 2: From this subset, Maximizer then selects the vertex with the **greatest** U(v) value. If more than one vertex meets these criteria, Maximizer can randomly select any one of these vertices.



Maximizer's Optimal Game-Play Strategy for Vertex Transitive Graphs

Step 1: Find all vertices with the **lowest** S(u) value.

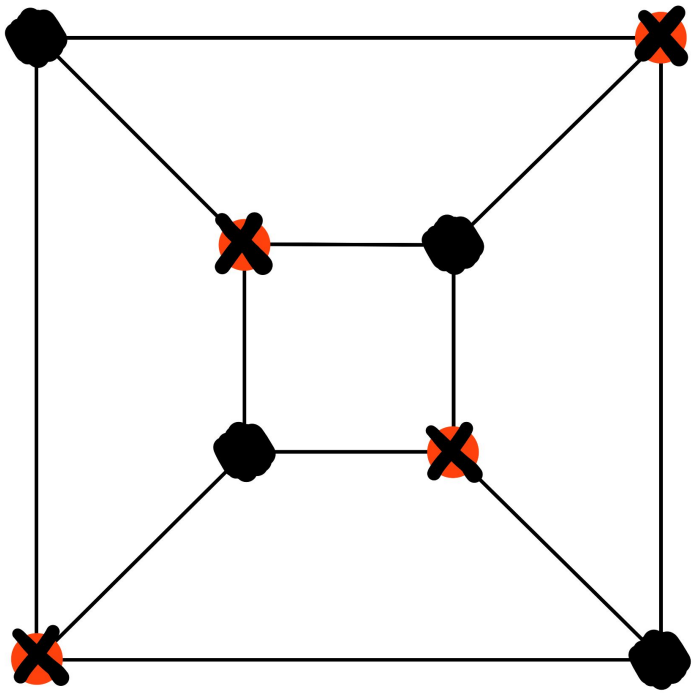
Step 2: From this subset, Maximizer then selects the vertex with the **greatest** U(v) value. If more than one vertex meets these criteria, Maximizer can randomly select any one of these vertices.



Maximizer's Optimal Game-Play Strategy for Vertex Transitive Graphs

Step 1: Find all vertices with the **lowest** S(u) value.

Step 2: From this subset, Maximizer then selects the vertex with the **greatest** U(v) value. If more than one vertex meets these criteria, Maximizer can randomly select any one of these vertices.

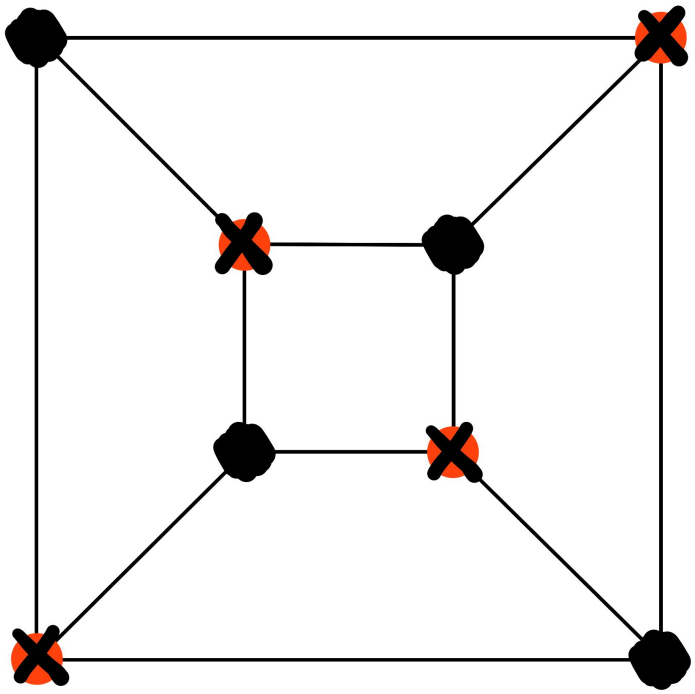


Maximizer's Optimal Game-Play Strategy for Vertex Transitive Graphs

Step 1: Find all vertices with the **lowest** S(u) value.

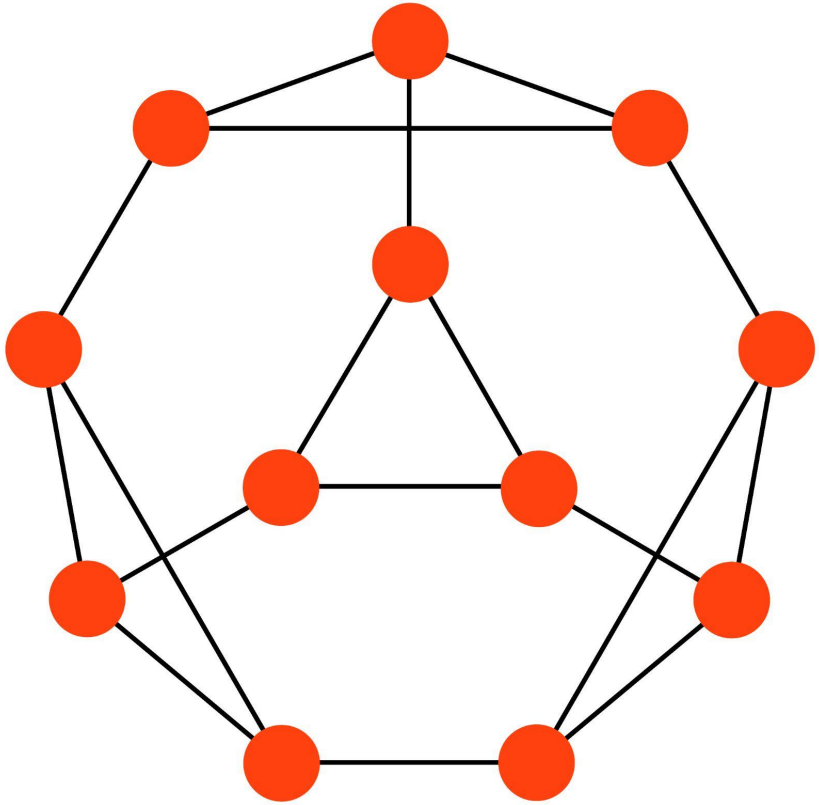
Step 2: From this subset, Maximizer then selects the vertex with the **greatest** U(v) value. If more than one vertex meets these criteria, Maximizer can randomly select any one of these vertices.

*The game-play strategies for Minimizer and Maximizer are developed for vertex-transitive graphs with a radius ≤ 3.



An Extra Example

An Extra Example



An Extra Example

Minimizer:

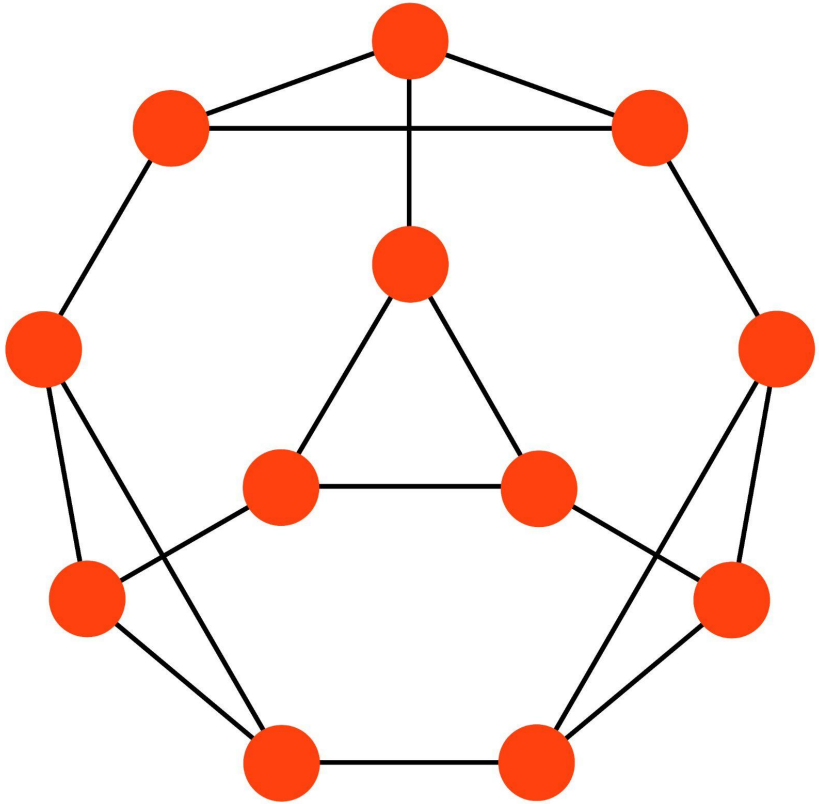
(1) Greatest <u>S(u) value</u>

(2) Lowest U(v) value

Maximizer:

(1) Lowest <u>S(u) value</u>

(2) Greatest U(v) value



An Extra Example

Minimizer:

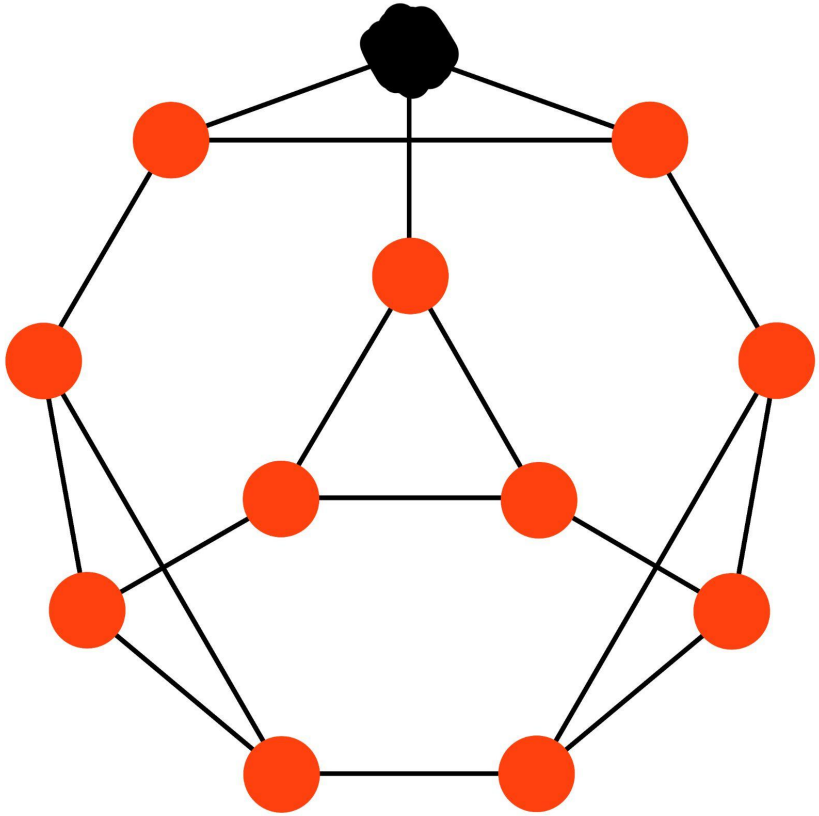
(1) Greatest <u>S(u) value</u>

(2) Lowest U(v) value

Maximizer:

(1) Lowest <u>S(u) value</u>

(2) Greatest U(v) value



An Extra Example

Minimizer:

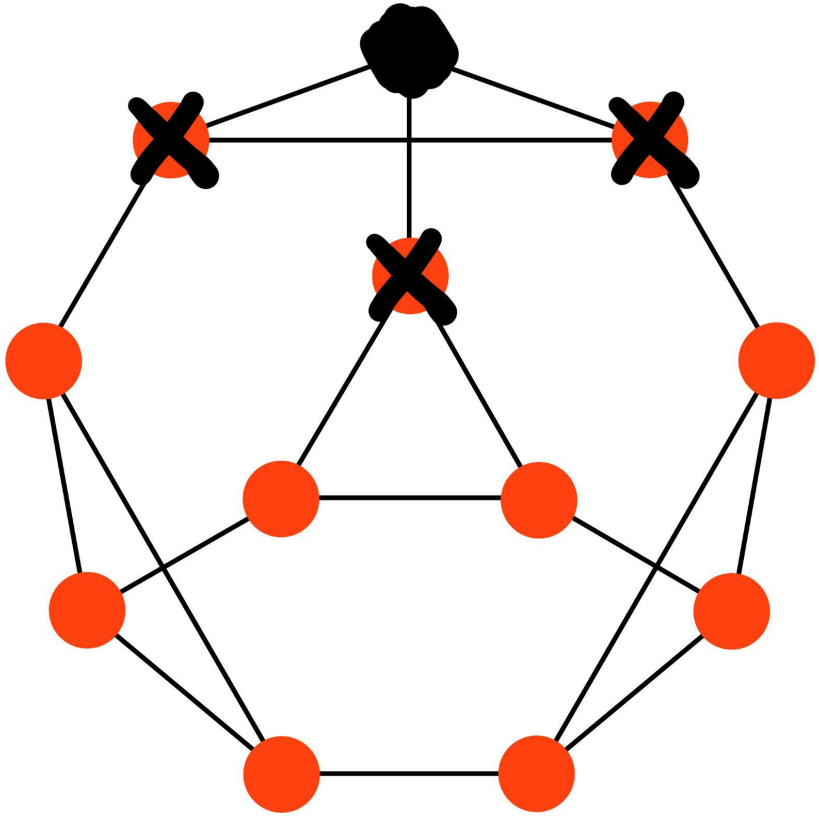
(1) Greatest <u>S(u) value</u>

(2) Lowest U(v) value

Maximizer:

(1) Lowest <u>S(u) value</u>

(2) Greatest U(v) value



An Extra Example

Minimizer:

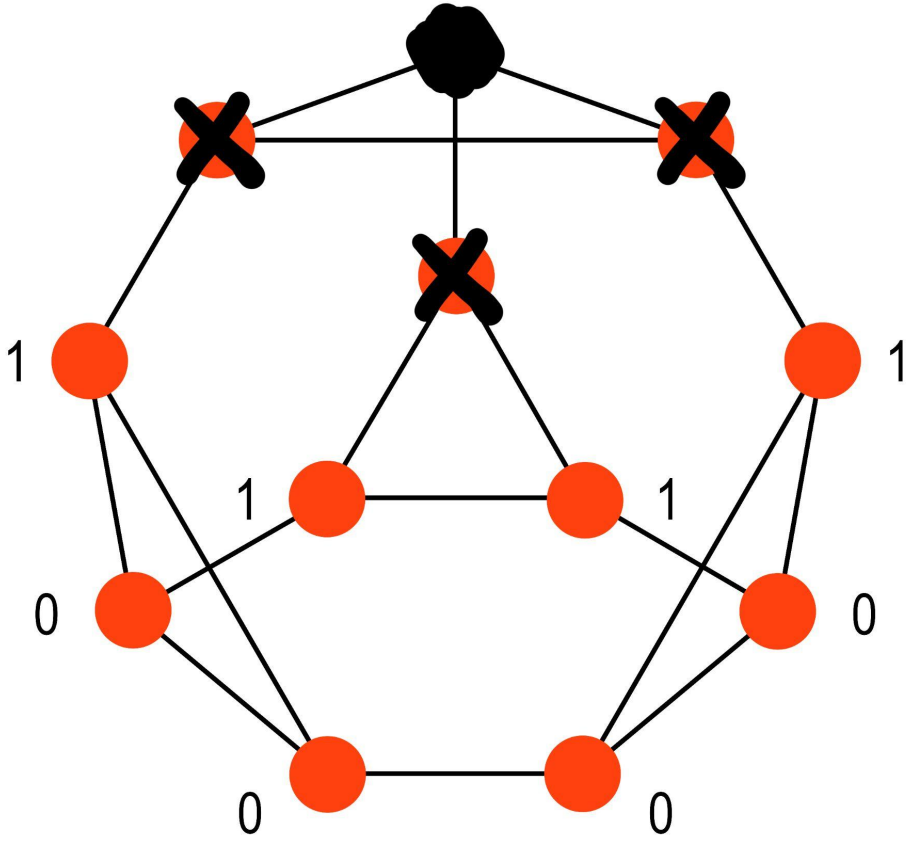
(1) Greatest <u>S(u) value</u>

(2) Lowest U(v) value

Maximizer:

(1) Lowest <u>S(u) value</u>

(2) Greatest U(v) value



An Extra Example

Minimizer:

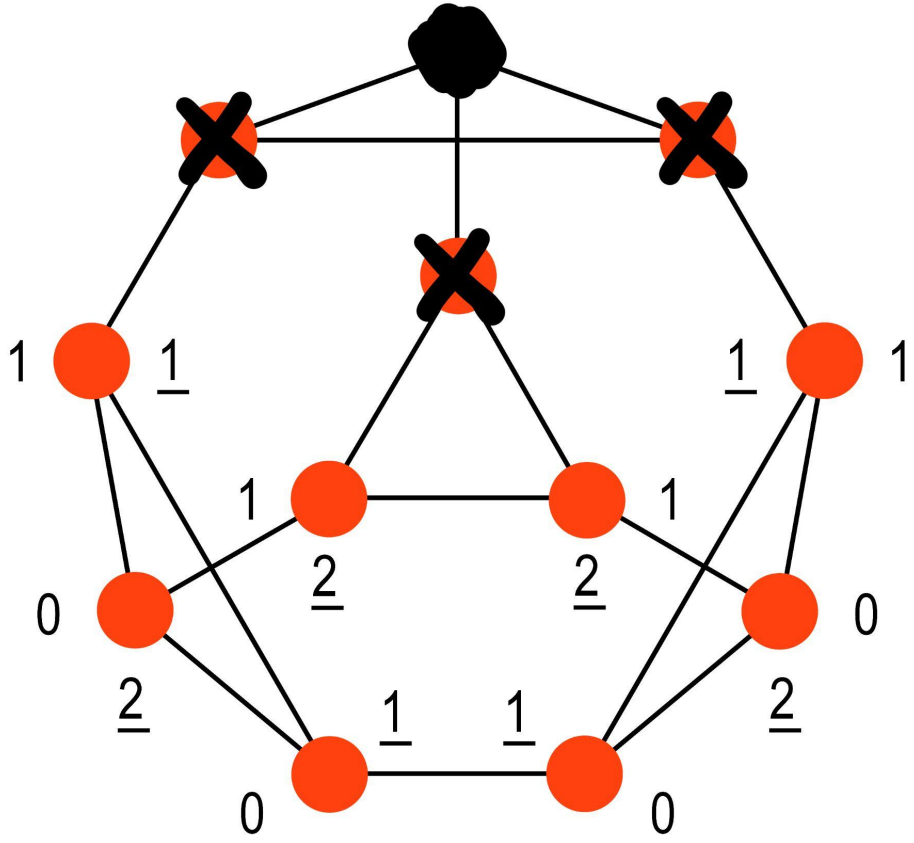
(1) Greatest <u>S(u) value</u>

(2) Lowest U(v) value

Maximizer:

(1) Lowest <u>S(u) value</u>

(2) Greatest U(v) value



An Extra Example

Minimizer:

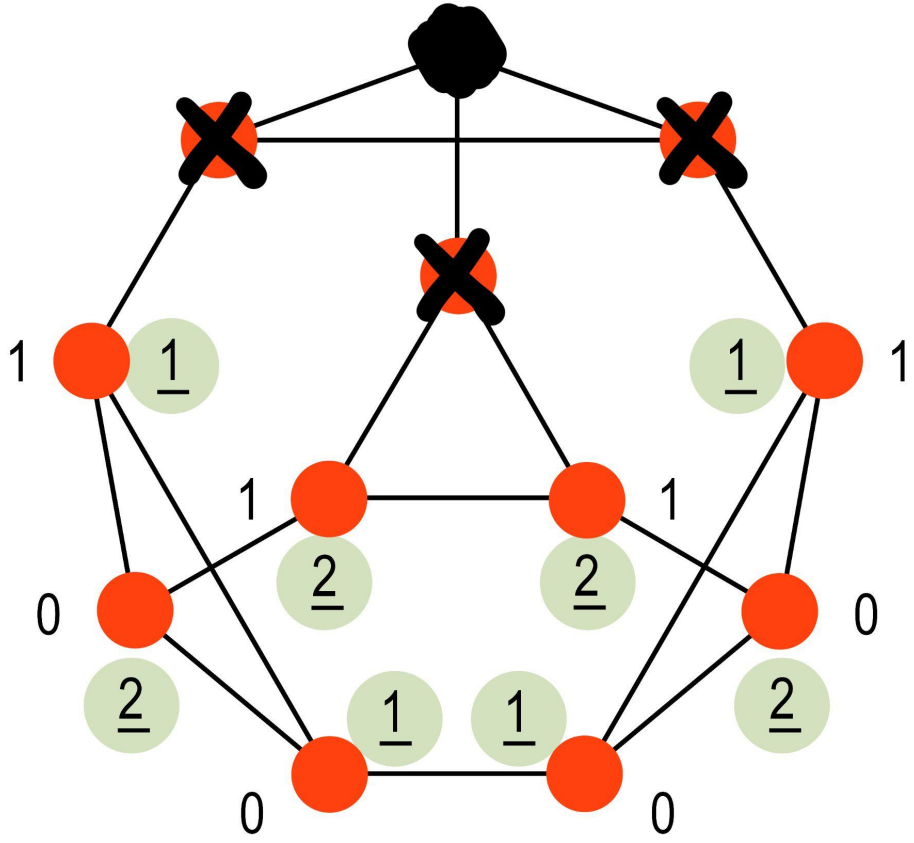
(1) Greatest <u>S(u) value</u>

(2) Lowest U(v) value

Maximizer:

(1) Lowest <u>S(u) value</u>

(2) Greatest U(v) value



An Extra Example

Minimizer:

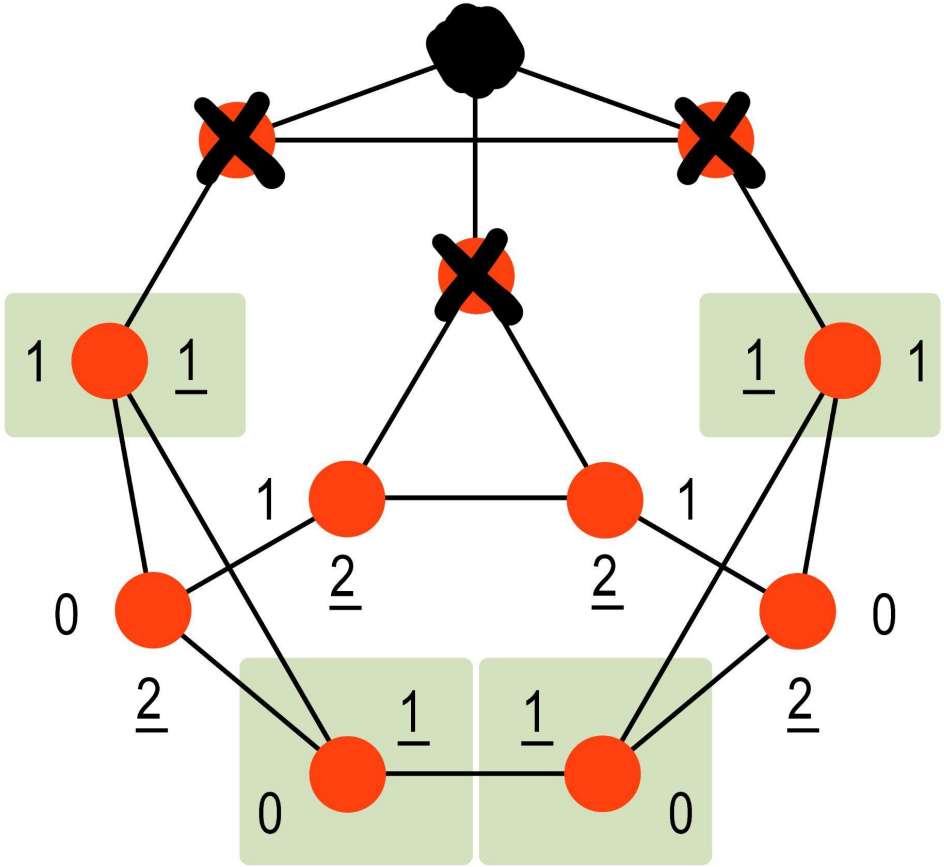
(1) Greatest <u>S(u) value</u>

(2) Lowest U(v) value

Maximizer:

(1) Lowest <u>S(u) value</u>

(2) Greatest U(v) value



An Extra Example

Minimizer:

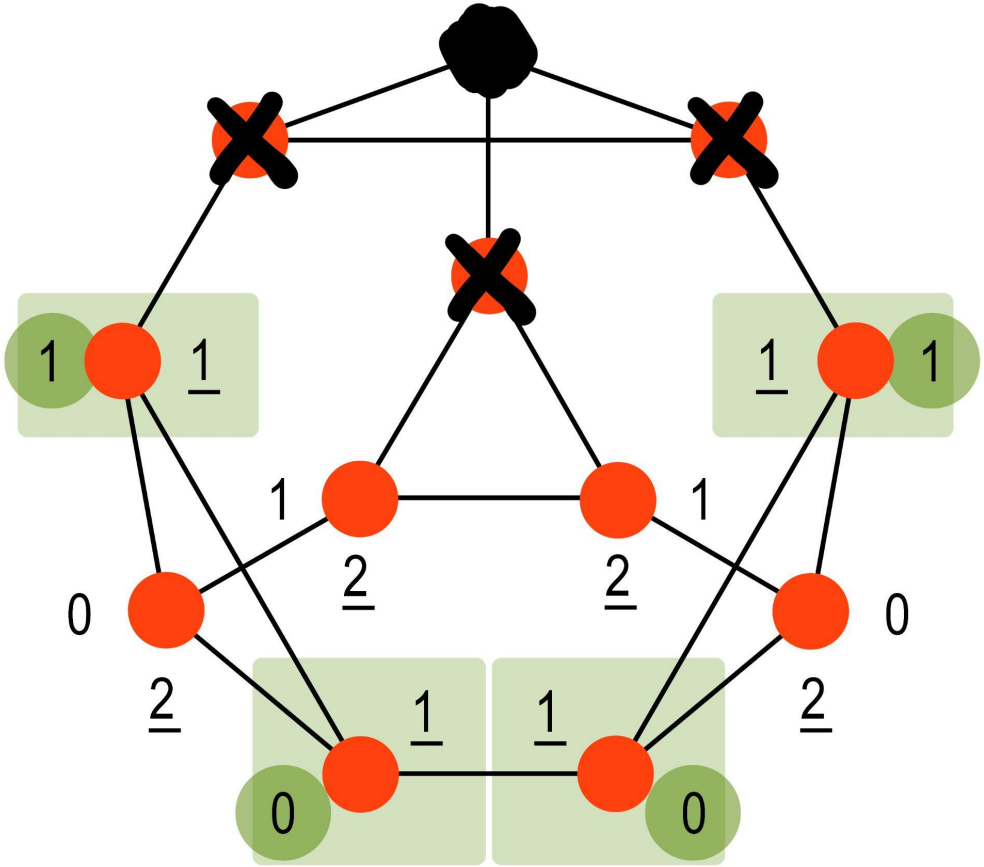
(1) Greatest <u>S(u) value</u>

(2) Lowest U(v) value

Maximizer:

(1) Lowest <u>S(u) value</u>

(2) Greatest U(v) value



An Extra Example

Minimizer:

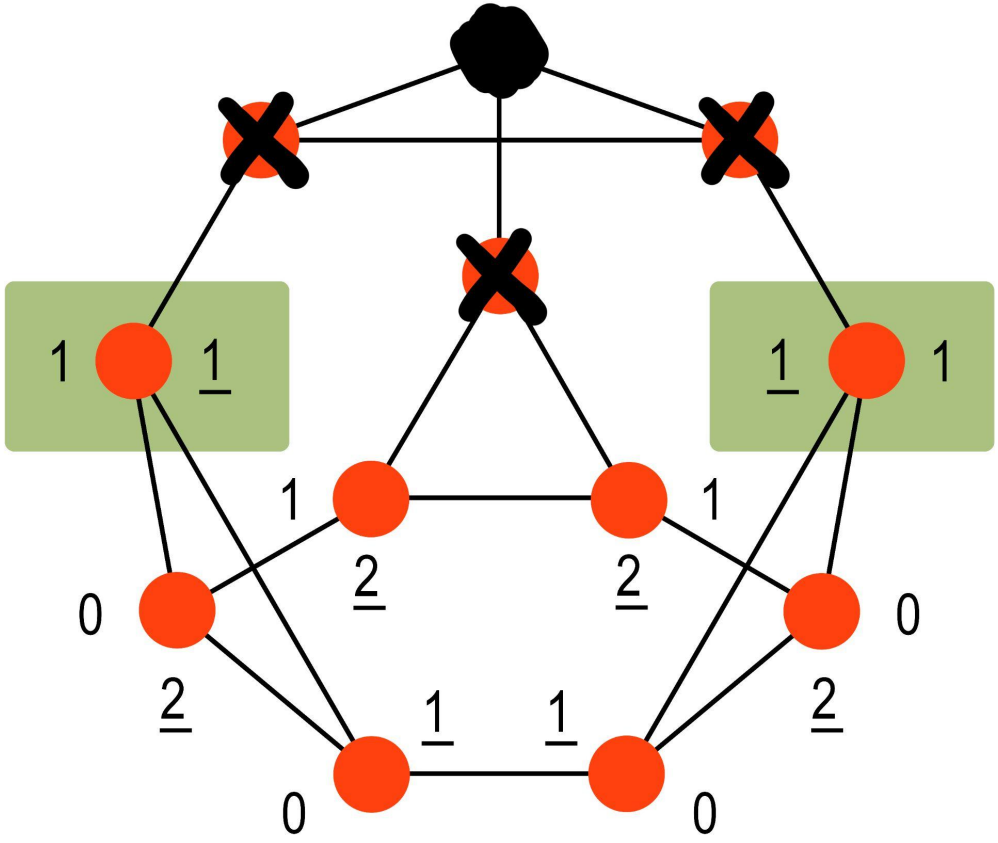
(1) Greatest <u>S(u) value</u>

(2) Lowest U(v) value

Maximizer:

(1) Lowest <u>S(u) value</u>

(2) Greatest U(v) value



An Extra Example

Minimizer:

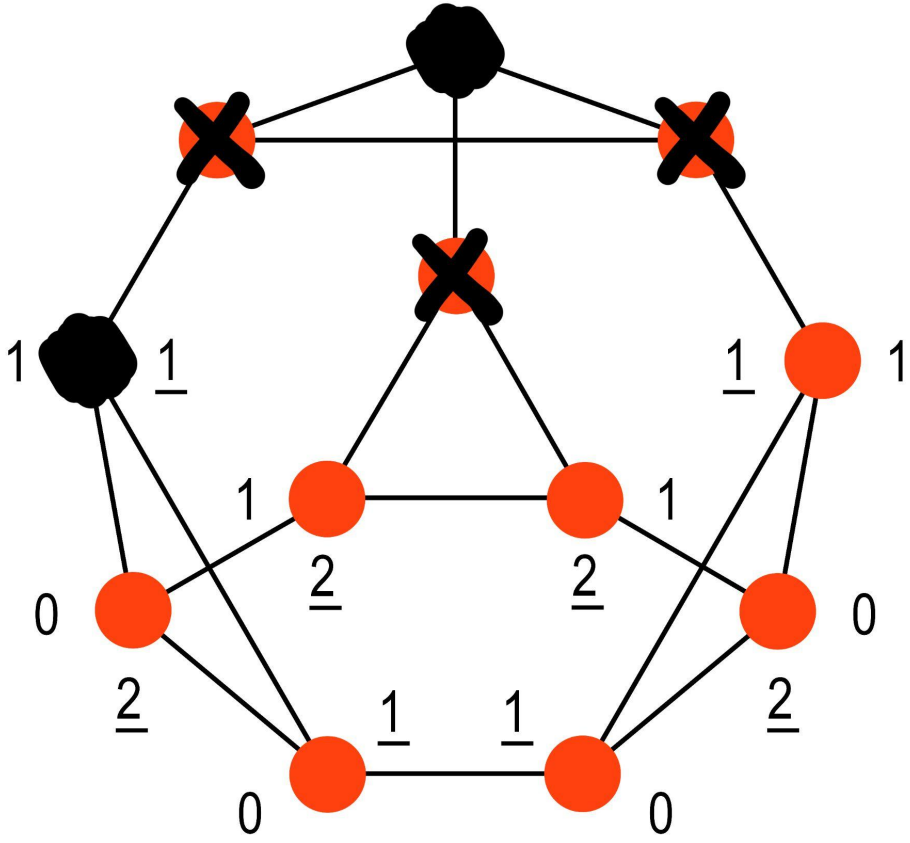
(1) Greatest <u>S(u) value</u>

(2) Lowest U(v) value

Maximizer:

(1) Lowest <u>S(u) value</u>

(2) Greatest U(v) value



An Extra Example

Minimizer:

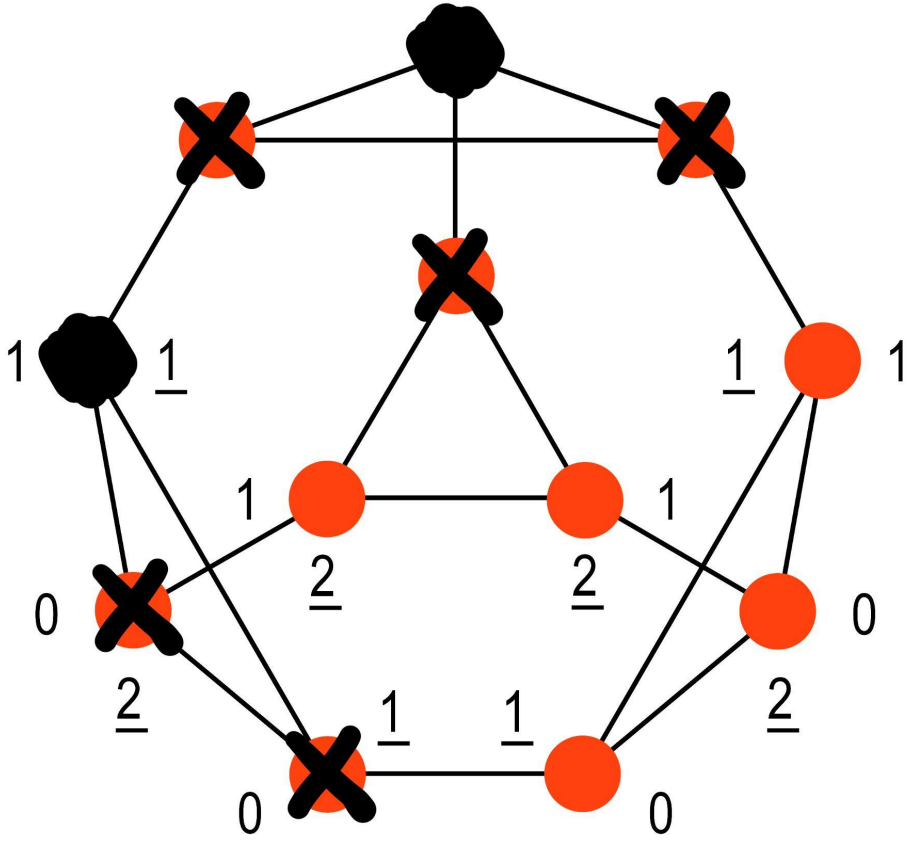
(1) Greatest <u>S(u) value</u>

(2) Lowest U(v) value

Maximizer:

(1) Lowest <u>S(u) value</u>

(2) Greatest U(v) value



An Extra Example

Minimizer:

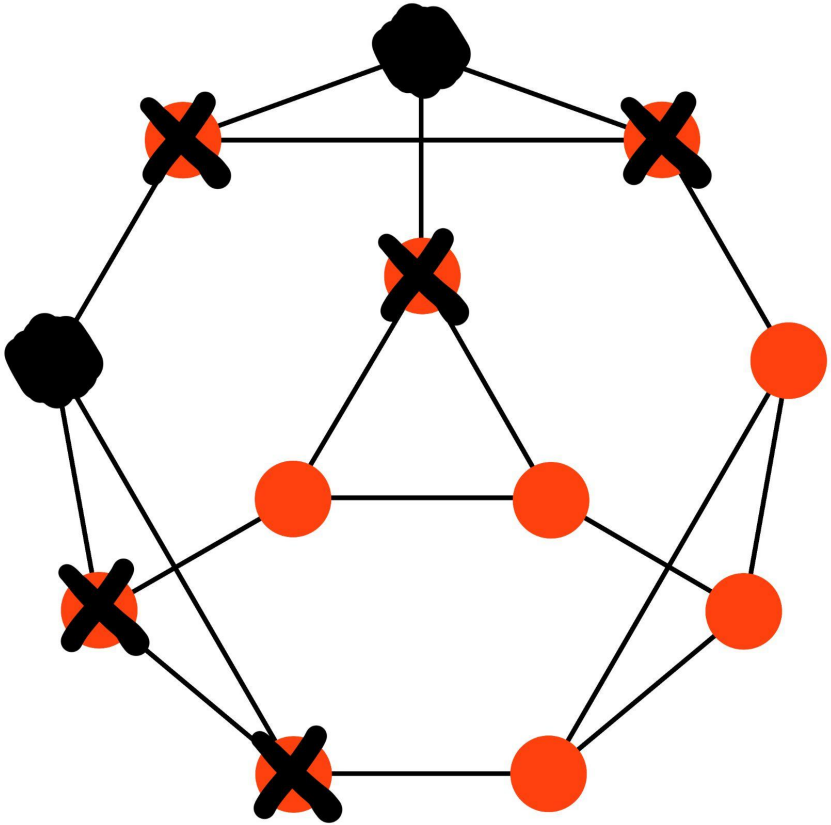
(1) Greatest <u>S(u) value</u>

(2) Lowest U(v) value

Maximizer:

(1) Lowest <u>S(u) value</u>

(2) Greatest U(v) value



An Extra Example

Minimizer:

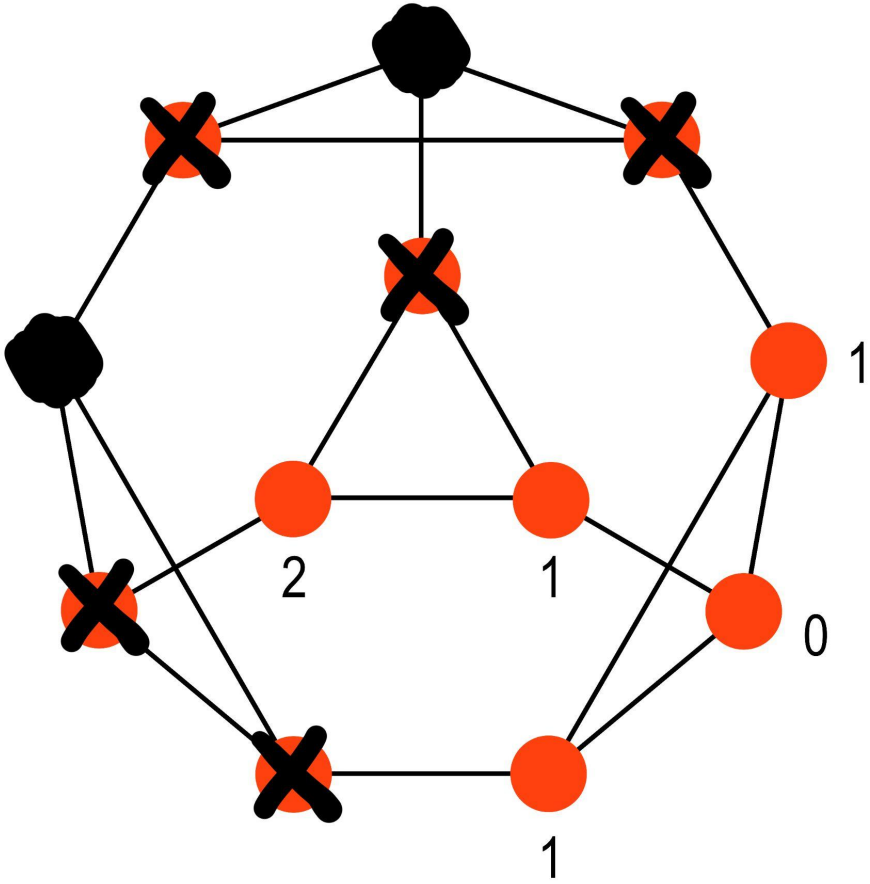
(1) Greatest <u>S(u) value</u>

(2) Lowest U(v) value

Maximizer:

(1) Lowest <u>S(u) value</u>

(2) Greatest U(v) value



An Extra Example

Minimizer:

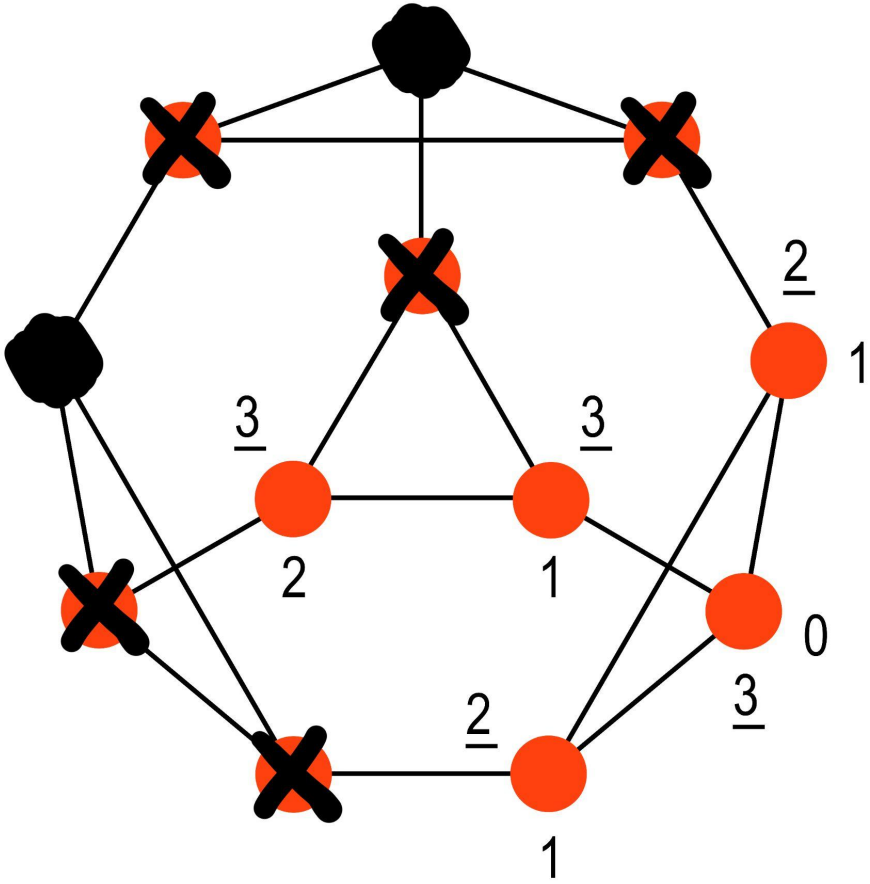
(1) Greatest <u>S(u) value</u>

(2) Lowest U(v) value

Maximizer:

(1) Lowest <u>S(u) value</u>

(2) Greatest U(v) value



An Extra Example

Minimizer:

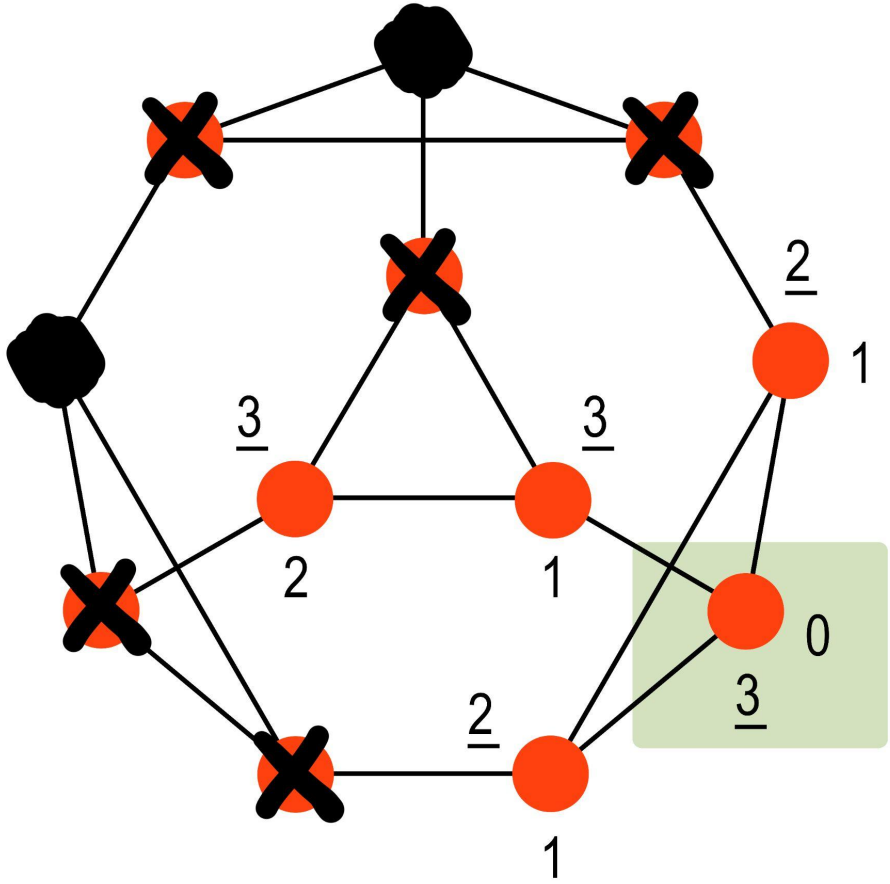
(1) Greatest <u>S(u) value</u>

(2) Lowest U(v) value

Maximizer:

(1) Lowest <u>S(u) value</u>

(2) Greatest U(v) value



An Extra Example

Minimizer:

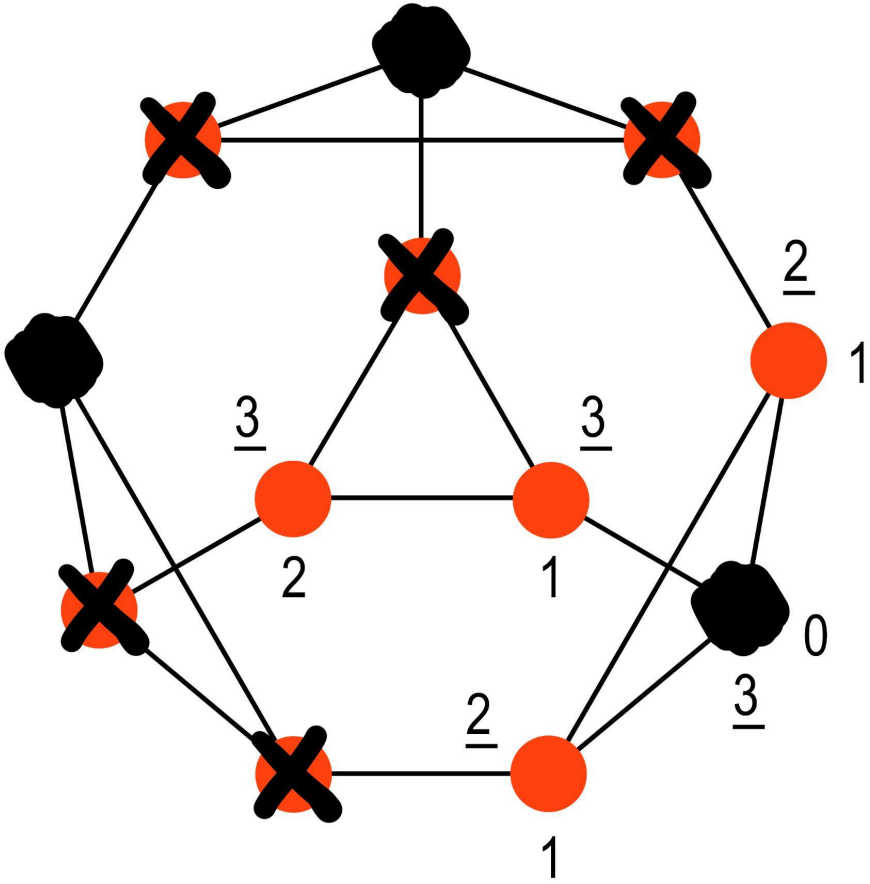
(1) Greatest <u>S(u) value</u>

(2) Lowest U(v) value

Maximizer:

(1) Lowest <u>S(u) value</u>

(2) Greatest U(v) value



An Extra Example

Minimizer:

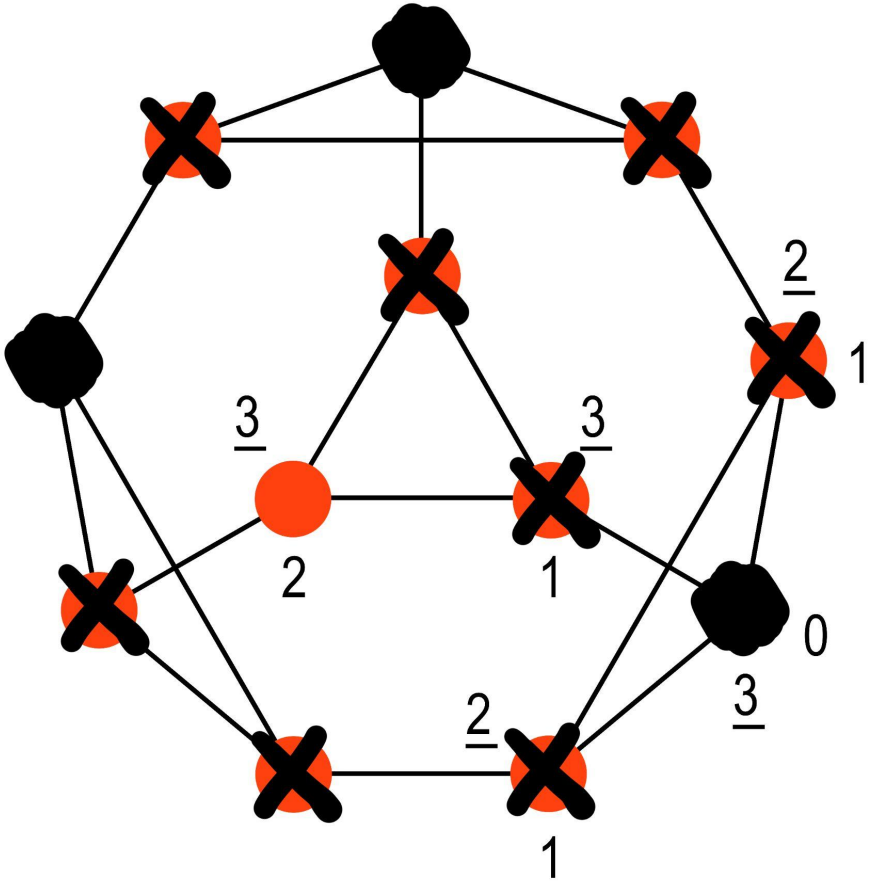
(1) Greatest <u>S(u) value</u>

(2) Lowest U(v) value

Maximizer:

(1) Lowest <u>S(u) value</u>

(2) Greatest U(v) value



An Extra Example

Minimizer:

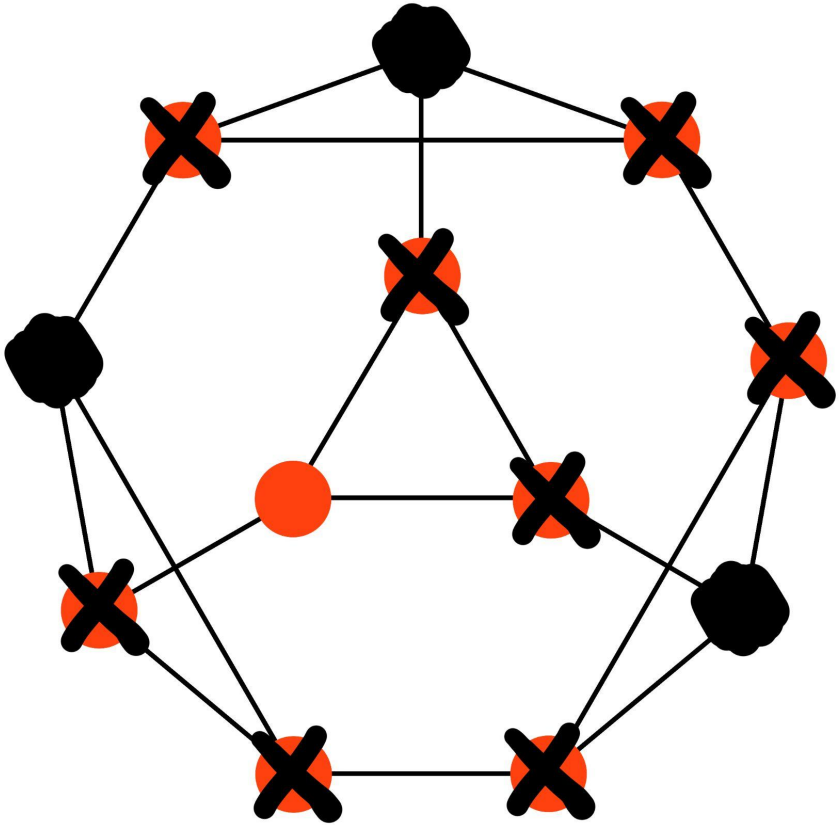
(1) Greatest <u>S(u) value</u>

(2) Lowest U(v) value

Maximizer:

(1) Lowest <u>S(u) value</u>

(2) Greatest U(v) value



An Extra Example

Minimizer:

(1) Greatest <u>S(u) value</u>

(2) Lowest U(v) value

Maximizer:

(1) Lowest <u>S(u) value</u>

(2) Greatest U(v) value