Graph Theory: Trees and Colorings

PRIMES Circle Program

Chloe Carrano and Eileen Lee

May 17, 2026

Introduction to Graph Theory

- Imagine that you and your friends are going to the cafeteria to eat lunch.

Introduction to Graph Theory

- Imagine that you and your friends are going to the cafeteria to eat lunch.
- You sit around a table, and you see that you know *some* of the people seated at the table but not all of them. Like you, everyone else only knows some of the people at the table.

Introduction to Graph Theory

- Imagine that you and your friends are going to the cafeteria to eat lunch.
- You sit around a table, and you see that you know *some* of the people seated at the table but not all of them. Like you, everyone else only knows some of the people at the table.
- You and your friends are named: Alice, Bonnie, Chloe, Dan, and Eileen.

Introduction to Graph Theory

- Imagine that you and your friends are going to the cafeteria to eat lunch.
- You sit around a table, and you see that you know *some* of the people seated at the table but not all of them. Like you, everyone else only knows some of the people at the table.
- You and your friends are named: Alice, Bonnie, Chloe, Dan, and Eileen.
- Let's try to find a seating arrangement where everyone sits next to someone they know.

Relationship Graph

- We can represent this with a graph!

Relationship Graph

- We can represent this with a graph!
- The vertices represent the people with names A-E, and the edges are drawn between two vertices if the people know each other.

Relationship Graph

- We can represent this with a graph!
- The vertices represent the people with names A-E, and the edges are drawn between two vertices if the people know each other.

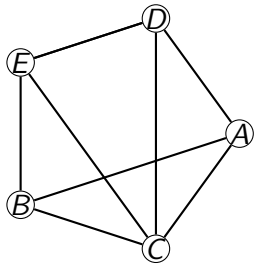


Figure: Graph representation

Relationship Graph

- We can represent this with a graph!
- The vertices represent the people with names A-E, and the edges are drawn between two vertices if the people know each other.

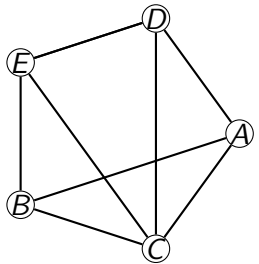


Figure: Graph representation

- The graph representation offers us a way to see the possible seatings.

Graph Basics

Now let's think about graphs more generally. We can formally define a graph as follows:

Graph Basics

Now let's think about graphs more generally. We can formally define a graph as follows:

Definition

A **graph** G consists of a finite nonempty set V of objects called vertices and a set E called edges of 2-element subsets of V.

Graph Basics

Now let's think about graphs more generally. We can formally define a graph as follows:

Definition

A **graph** G consists of a finite nonempty set V of objects called vertices and a set E called edges of 2-element subsets of V.

Every vertex on a graph is incident to a certain number of edges. We refer to this number of edges as a vertex's degree.

Graph Basics

Now let's think about graphs more generally. We can formally define a graph as follows:

Definition

A **graph** G consists of a finite nonempty set V of objects called vertices and a set E called edges of 2-element subsets of V.

Every vertex on a graph is incident to a certain number of edges. We refer to this number of edges as a vertex's degree.

Definition

The **degree** of a vertex v, denoted $\deg(v)$, is the number of edges incident with v.

Tree Graphs

with

Applications to Computer Science

Chloe Carrano

Tree Structures

Definition

A **tree** is an acyclic, connected graph.

Tree Structures

Definition

A **tree** is an acyclic, connected graph.

2 parts to the definition of a tree: being (1) acyclic, and (2) connected. Let's look at each part individually.

Tree Structures

Definition

A **tree** is an acyclic, connected graph.

2 parts to the definition of a tree: being (1) acyclic, and (2) connected. Let's look at each part individually.

Definition

A graph is **connected** if for any two vertices x, y within $V(G)$, there is a path whose endpoints are x and y.

Tree Structures

Definition

A **tree** is an acyclic, connected graph.

2 parts to the definition of a tree: being (1) acyclic, and (2) connected. Let's look at each part individually.

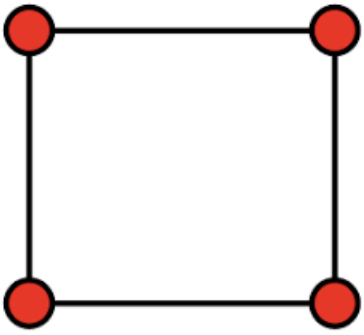
Definition

A graph is **connected** if for any two vertices x, y within $V(G)$, there is a path whose endpoints are x and y.

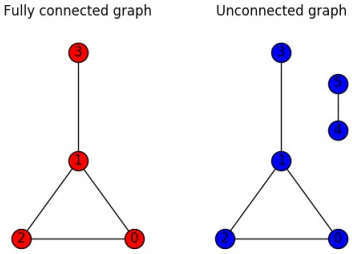
Definition

A graph is described to be **acyclic** if it contains no cycles.

Cycles and Connected Graphs



(a) A cycle with 4 vertices (4-cycle)

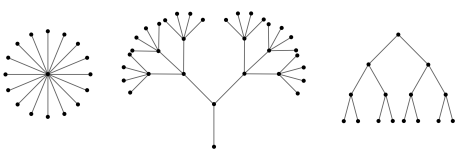


(b) Two graphs showing the definition of connectedness

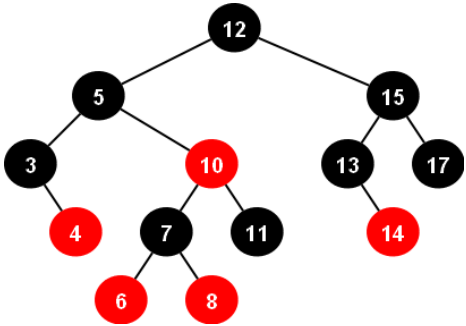
We represent a cycle graph with n vertices ($n \geq 3$) connected in a single closed chain with the notation C_n.

Trees

Again, tree graphs must be connected and contain no cycles! They may be colored, weighted, or sorted by vertex value (more on that later...)

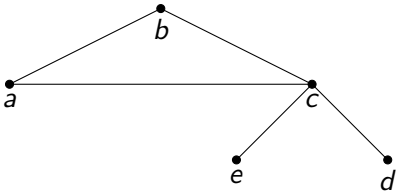


(a) Examples of trees – we would call these (as an entire group) a forest.

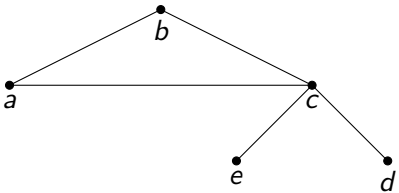


(b) A particular type of tree (red-black tree) with colored, weighted vertices.

Spanning Trees

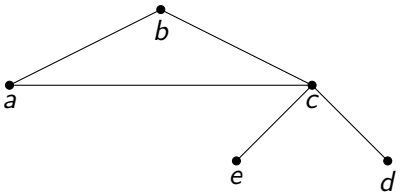


Spanning Trees



- Imagine you're in charge of designing a new network of train lines for the MBTA connecting five cities: Agria, Biloti, Clodo, Durley, and Emmio.

Spanning Trees



- Imagine you're in charge of designing a new network of train lines for the MBTA connecting five cities: Agria, Biloti, Clodo, Durley, and Emmio.
- One day, a tree falls on the train tracks between Biloti and Clodo. It looks like we can still get from any one city to another, even with this broken track! However, with any second, additional track broken, that wouldn't be possible.

Spanning Trees, Continued

- With B to C gone, we now have a **spanning tree**.

Spanning Trees, Continued

- With B to C gone, we now have a **spanning tree**. Official definition of a spanning tree:

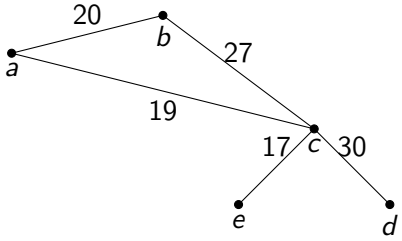
Spanning Trees, Continued

- With B to C gone, we now have a **spanning tree**. Official definition of a spanning tree:

Definition

A spanning subgraph H of a connected graph G such that H is a tree is called a **spanning tree** of G.

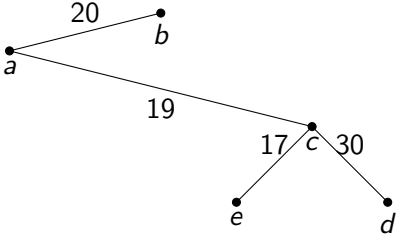
- If we want to minimize the cost of building entirely new train tracks, we can weight each edge on the graph with the cost it might take to rebuild:



Spanning Trees, Continued

- To minimize costs, we want to find the **minimum spanning tree** within this graph.
- We can use several different algorithms to find the minimum spanning tree of a weighted graph, but for now we'll use Kruskal's algorithm.

Using Kruskal's, we'll continuously select the lowest-weight available edge that connects two disconnected components (while ensuring no cycles are formed) until all vertices are connected. We're left with the following graph:



Kirchhoff's Matrix Tree Theorem

Gustav Kirchhoff, a German physicist and revolutionary electrical engineer, developed a theorem involving matrices to determine the total number of distinct spanning trees in a given graph – we call this Kirchhoff's Matrix Tree Theorem.

Theorem (Kirchhoff's Matrix-Tree Theorem)

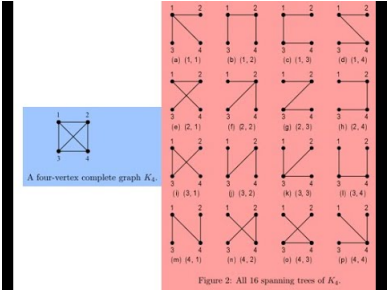
The number of spanning trees in a graph G is given by $\det(L_G[i])$ for any i.

Kirchhoff's Matrix Tree Theorem

Gustav Kirchhoff, a German physicist and revolutionary electrical engineer, developed a theorem involving matrices to determine the total number of distinct spanning trees in a given graph – we call this Kirchhoff's Matrix Tree Theorem.

Theorem (Kirchhoff's Matrix-Tree Theorem)

The number of spanning trees in a graph G is given by $\det(L_G[i])$ for any i.



Tree Variations

Besides spanning trees, other tree variations include:

Tree Variations

Besides spanning trees, other tree variations include:

- Binary search trees (each edge comes from a single **root** point and branches out to form ≤ 2 other edges)

Tree Variations

Besides spanning trees, other tree variations include:

- Binary search trees (each edge comes from a single **root** point and branches out to form ≤ 2 other edges)
- Minimum and maximum heaps (trees where every **node** has a specific value – root has the min or max value)

Tree Variations

Besides spanning trees, other tree variations include:

- Binary search trees (each edge comes from a single **root** point and branches out to form ≤ 2 other edges)
- Minimum and maximum heaps (trees where every **node** has a specific value – root has the min or max value)
- Huffman trees (only store data at particular vertices and minimize the weight sum of those vertices)

Tree Variations

Besides spanning trees, other tree variations include:

- Binary search trees (each edge comes from a single **root** point and branches out to form ≤ 2 other edges)
- Minimum and maximum heaps (trees where every **node** has a specific value – root has the min or max value)
- Huffman trees (only store data at particular vertices and minimize the weight sum of those vertices)
- Prefix trees (each vertex represents a particular character)

Tree Variations

Besides spanning trees, other tree variations include:

- Binary search trees (each edge comes from a single **root** point and branches out to form ≤ 2 other edges)
- Minimum and maximum heaps (trees where every **node** has a specific value – root has the min or max value)
- Huffman trees (only store data at particular vertices and minimize the weight sum of those vertices)
- Prefix trees (each vertex represents a particular character)

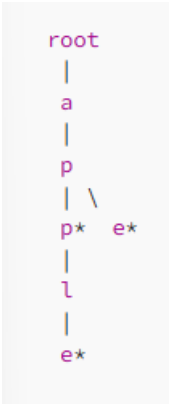


Figure: A prefix tree storing the strings "app," "ape," and "apple."

Applications of Trees

Tree structures have been in use since antiquity, first created by medieval clerics to represent potential matchmaking relationships, then later used within religious contexts to form the famous 11th-century Tree of Jesse and show connections between the Greek classical gods.

Applications of Trees

Tree structures have been in use since antiquity, first created by medieval clerics to represent potential matchmaking relationships, then later used within religious contexts to form the famous 11th-century Tree of Jesse and show connections between the Greek classical gods. Modern uses:

Applications of Trees

Tree structures have been in use since antiquity, first created by medieval clerics to represent potential matchmaking relationships, then later used within religious contexts to form the famous 11th-century Tree of Jesse and show connections between the Greek classical gods. Modern uses:

- Genealogy (representing ancestry)

Applications of Trees

Tree structures have been in use since antiquity, first created by medieval clerics to represent potential matchmaking relationships, then later used within religious contexts to form the famous 11th-century Tree of Jesse and show connections between the Greek classical gods. Modern uses:

- Genealogy (representing ancestry)
- Chemistry (representing molecular structures)

Applications of Trees

Tree structures have been in use since antiquity, first created by medieval clerics to represent potential matchmaking relationships, then later used within religious contexts to form the famous 11th-century Tree of Jesse and show connections between the Greek classical gods. Modern uses:

- Genealogy (representing ancestry)
- Chemistry (representing molecular structures)
- Taxonomy (representing hypothetical relationships between organisms within a particular classification, such as a phylum)

Applications of Trees

Tree structures have been in use since antiquity, first created by medieval clerics to represent potential matchmaking relationships, then later used within religious contexts to form the famous 11th-century Tree of Jesse and show connections between the Greek classical gods. Modern uses:

- Genealogy (representing ancestry)
- Chemistry (representing molecular structures)
- Taxonomy (representing hypothetical relationships between organisms within a particular classification, such as a phylum)
- Data management

Applications of Trees

Tree structures have been in use since antiquity, first created by medieval clerics to represent potential matchmaking relationships, then later used within religious contexts to form the famous 11th-century Tree of Jesse and show connections between the Greek classical gods. Modern uses:

- Genealogy (representing ancestry)
- Chemistry (representing molecular structures)
- Taxonomy (representing hypothetical relationships between organisms within a particular classification, such as a phylum)
- Data management

Tree structures are everywhere – the next time that you see one in your daily life, think of the graph theory concepts behind their construction and properties!

Coloring Graphs

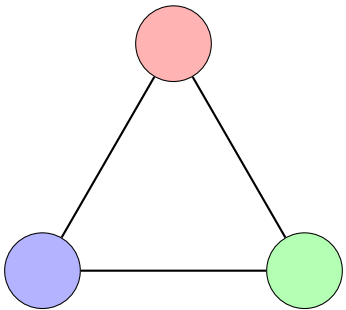
and

Ramsey Numbers

Eileen Lee

Part I

Graph Colorings



Vertex Coloring, Edge Coloring, and Applications

Vertex Coloring

Definition

A **proper vertex coloring** assigns colors to vertices so that adjacent vertices receive different colors.

Vertex Coloring

Definition

A **proper vertex coloring** assigns colors to vertices so that adjacent vertices receive different colors.

Definition

The minimum number of colors needed is called the **chromatic number**:

$$\chi(G)$$

Vertex Coloring

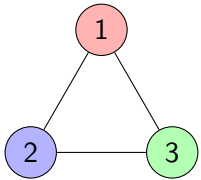
Definition

A **proper vertex coloring** assigns colors to vertices so that adjacent vertices receive different colors.

Definition

The minimum number of colors needed is called the **chromatic number**:

$$\chi(G)$$



Vertex Coloring

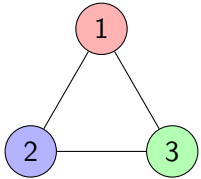
Definition

A **proper vertex coloring** assigns colors to vertices so that adjacent vertices receive different colors.

Definition

The minimum number of colors needed is called the **chromatic number**:

$$\chi(G)$$



$$\chi(C_3) = 3$$

Bipartite Graphs

Definition

A graph is **bipartite** if its vertices can be divided into two sets so that every edge connects vertices from different sets.

Bipartite Graphs

Definition

A graph is **bipartite** if its vertices can be divided into two sets so that every edge connects vertices from different sets.

Theorem

A nonempty graph with a nonempty edge set is bipartite if and only if it is 2-colorable.

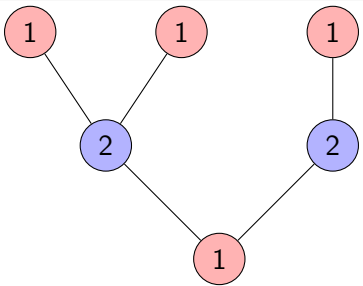
Bipartite Graphs

Definition

A graph is **bipartite** if its vertices can be divided into two sets so that every edge connects vertices from different sets.

Theorem

A nonempty graph with a nonempty edge set is bipartite if and only if it is 2-colorable.



Important Coloring Result

Theorem

For every graph G,

$$\chi(G) \leq \Delta(G) + 1$$

where $\Delta(G)$ is the maximum degree of the graph.

Important Coloring Result

Theorem

For every graph G,

$$\chi(G) \leq \Delta(G) + 1$$

where $\Delta(G)$ is the maximum degree of the graph.

- A vertex can have at most $\Delta(G)$ neighbors.

Important Coloring Result

Theorem

For every graph G,

$$\chi(G) \leq \Delta(G) + 1$$

where $\Delta(G)$ is the maximum degree of the graph.

- A vertex can have at most $\Delta(G)$ neighbors.
- Therefore at most $\Delta(G)$ colors are forbidden.

Important Coloring Result

Theorem

For every graph G,

$$\chi(G) \leq \Delta(G) + 1$$

where $\Delta(G)$ is the maximum degree of the graph.

- A vertex can have at most $\Delta(G)$ neighbors.
- Therefore at most $\Delta(G)$ colors are forbidden.
- One additional color is always enough.

Important Coloring Result

Theorem

For every graph G,

$$\chi(G) \leq \Delta(G) + 1$$

where $\Delta(G)$ is the maximum degree of the graph.

- A vertex can have at most $\Delta(G)$ neighbors.
- Therefore at most $\Delta(G)$ colors are forbidden.
- One additional color is always enough.

This gives an upper bound on the chromatic number.

Scheduling Applications

- Graph coloring can solve scheduling problems.

Scheduling Applications

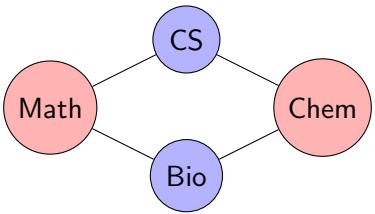
- Graph coloring can solve scheduling problems.
- Vertices represent courses (such as math) that are available in the school.

Scheduling Applications

- Graph coloring can solve scheduling problems.
- Vertices represent courses (such as math) that are available in the school.
- Edges represent scheduling conflicts where a student wants to take both courses.

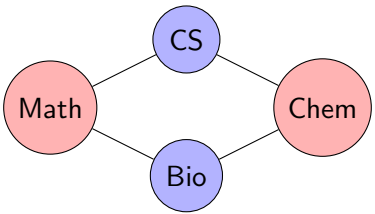
Scheduling Applications

- Graph coloring can solve scheduling problems.
- Vertices represent courses (such as math) that are available in the school.
- Edges represent scheduling conflicts where a student wants to take both courses.



Scheduling Applications

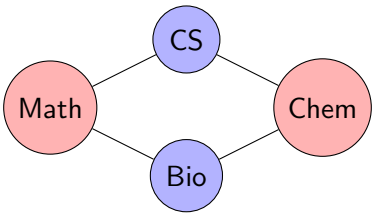
- Graph coloring can solve scheduling problems.
- Vertices represent courses (such as math) that are available in the school.
- Edges represent scheduling conflicts where a student wants to take both courses.



- Adjacent classes cannot be scheduled at the same time.

Scheduling Applications

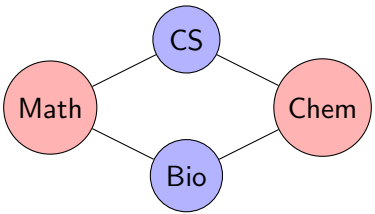
- Graph coloring can solve scheduling problems.
- Vertices represent courses (such as math) that are available in the school.
- Edges represent scheduling conflicts where a student wants to take both courses.



- Adjacent classes cannot be scheduled at the same time.
- Colors correspond to time slots.

Scheduling Applications

- Graph coloring can solve scheduling problems.
- Vertices represent courses (such as math) that are available in the school.
- Edges represent scheduling conflicts where a student wants to take both courses.



- Adjacent classes cannot be scheduled at the same time.
- Colors correspond to time slots.
- Only 2 time slots are needed here.

Edge Coloring

Definition

An **edge coloring** assigns colors to edges so that adjacent edges receive different colors.

Edge Coloring

Definition

An **edge coloring** assigns colors to edges so that adjacent edges receive different colors.

Definition

The minimum number of edge colors needed is called the **chromatic index**:

$$\chi'(G)$$

Edge Coloring

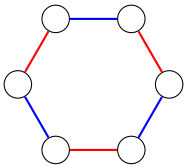
Definition

An **edge coloring** assigns colors to edges so that adjacent edges receive different colors.

Definition

The minimum number of edge colors needed is called the **chromatic index**:

$$\chi'(G)$$



Edge Coloring

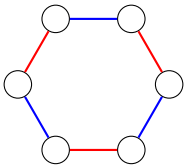
Definition

An **edge coloring** assigns colors to edges so that adjacent edges receive different colors.

Definition

The minimum number of edge colors needed is called the **chromatic index**:

$$\chi'(G)$$



$$\chi'(C_6) = 2$$

Vizing's Theorem

Vizing's Theorem

For every graph G,

$$\Delta(G) \leq \chi'(G) \leq \Delta(G) + 1$$

Vizing's Theorem

Vizing's Theorem

For every graph G,

$$\Delta(G) \leq \chi'(G) \leq \Delta(G) + 1$$

- Every graph needs at least $\Delta(G)$ edge colors.

Vizing's Theorem

Vizing's Theorem

For every graph G,

$$\Delta(G) \leq \chi'(G) \leq \Delta(G) + 1$$

- Every graph needs at least $\Delta(G)$ edge colors.
- At most one additional color is ever needed.

Vizing's Theorem

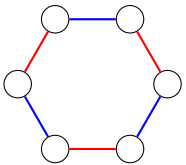
Vizing's Theorem

For every graph G,

$$\Delta(G) \leq \chi'(G) \leq \Delta(G) + 1$$

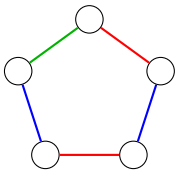
- Every graph needs at least $\Delta(G)$ edge colors.
- At most one additional color is ever needed.

Even Cycle



$\chi'(C_6) = 2$

Odd Cycle



$\chi'(C_5) = 3$

Part II

Ramsey Numbers

Ramsey Theory

- Ramsey theory studies unavoidable patterns.

Ramsey Theory

- Ramsey theory studies unavoidable patterns.
- Complete disorder is impossible in large systems.

Ramsey Theory

- Ramsey theory studies unavoidable patterns.
- Complete disorder is impossible in large systems.

Ramsey Number

$r(F_1, F_2)$ is the smallest integer n such that every red-blue coloring (of edges!) of the complete graph K_n contains:

- a red copy of graph F_1, or
- a blue copy of graph F_2

Ramsey Theory

- Ramsey theory studies unavoidable patterns.
- Complete disorder is impossible in large systems.

Ramsey Number

$r(F_1, F_2)$ is the smallest integer n such that every red-blue coloring (of edges!) of the complete graph K_n contains:

- a red copy of graph F_1, or
- a blue copy of graph F_2

- K_n is a graph with n vertices where every possible edge is drawn.

The Famous Result $r(K_3, K_3) = 6$

Theorem

$$r(K_3, K_3) = 6$$

The Famous Result $r(K_3, K_3) = 6$

Theorem

$$r(K_3, K_3) = 6$$

- K_3 is simply a triangle.

The Famous Result $r(K_3, K_3) = 6$

Theorem

$$r(K_3, K_3) = 6$$

- K_3 is simply a triangle.
- If we color every edge of a complete graph with 6 vertices red or blue, there must always be either a red triangle or a blue triangle.

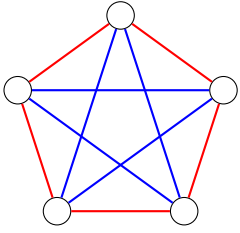
The Famous Result $r(K_3, K_3) = 6$

Theorem

$$r(K_3, K_3) = 6$$

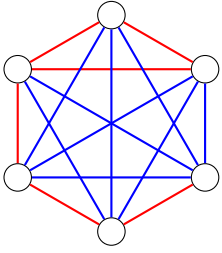
- K_3 is simply a triangle.
- If we color every edge of a complete graph with 6 vertices red or blue, there must always be either a red triangle or a blue triangle.

K_5



Avoids monochromatic triangles

K_6



A monochromatic triangle appears

Protein–Protein Interaction Networks

Protein–protein interaction networks model how proteins inside a cell communicate and work together.

- Vertices represent proteins

Protein–Protein Interaction Networks

Protein–protein interaction networks model how proteins inside a cell communicate and work together.

- Vertices represent proteins
- Edges represent interactions between proteins

Protein–Protein Interaction Networks

Protein–protein interaction networks model how proteins inside a cell communicate and work together.

- Vertices represent proteins
- Edges represent interactions between proteins

These networks help biologists study how groups of proteins coordinate important cellular functions.

Proteins in the Model

We consider several proteins involved in cell regulation and signaling:

- *TP53*
- *MDM2*
- *RB1*
- *CDK2*
- *EGFR*
- *GRB2*
- *SOS1*

Interaction Types in the Network

Interactions are simplified into two categories:

- **Strong interactions**: stable functional relationships

Interaction Types in the Network

Interactions are simplified into two categories:

- **Strong interactions**: stable functional relationships
- **Weak interactions**: temporary or signaling-based relationships

Protein–Protein Interaction Network

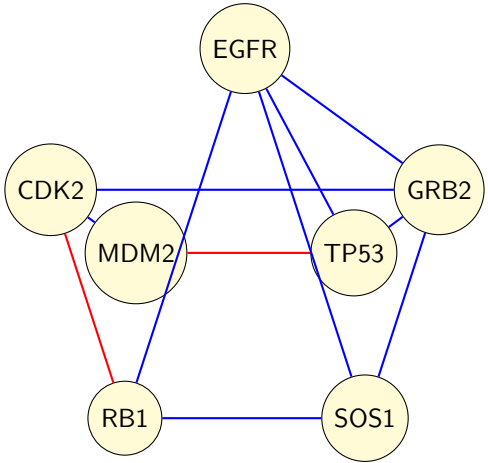


Figure: Red edges represent strong interactions, while blue edges represent weaker signaling interactions.

Ramsey Theory in Biological Networks

- Although the network appears irregular at first glance, Ramsey theory implies that complete disorder cannot persist in sufficiently large systems.

Ramsey Theory in Biological Networks

- Although the network appears irregular at first glance, Ramsey theory implies that complete disorder cannot persist in sufficiently large systems.
- Eventually, subsets of proteins must appear in which interactions become uniform.

Protein–Protein Interaction Network

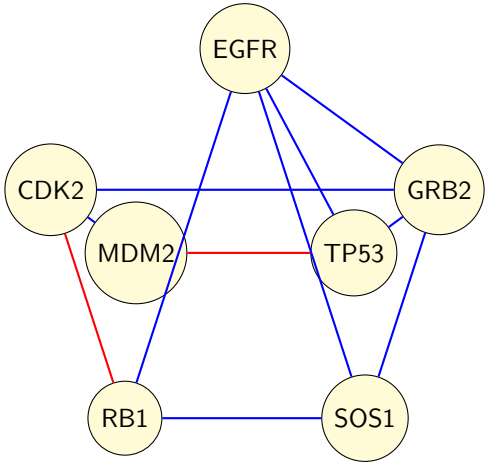


Figure: Red edges represent strong interactions, while blue edges represent weaker signaling interactions.

Ramsey Theory in Biological Networks

- These structures correspond biologically to:
  - stable protein complexes
  - signaling pathways
  - functional cellular modules

Ramsey Theory in Biological Networks

- These structures correspond biologically to:
  - stable protein complexes
  - signaling pathways
  - functional cellular modules
- Ramsey theory therefore provides a mathematical explanation for why biological systems naturally organize into clusters and coordinated pathways.

Conclusion

- Graph theory studies relationships and networks.

Conclusion

- Graph theory studies relationships and networks.
- Graph coloring can be used for solving optimization problems.

Conclusion

- Graph theory studies relationships and networks.
- Graph coloring can be used for solving optimization problems.
- Ramsey theory shows patterns inevitably emerge.

Conclusion

- Graph theory studies relationships and networks.
- Graph coloring can be used for solving optimization problems.
- Ramsey theory shows patterns inevitably emerge.
- These ideas connect mathematics with:
  - computer science
  - biology
  - social sciences
  - etc

Questions?