Graph Theory OS: Master Test

System Admin

1 1. Mathematical Notation

We test the semantic aliases to ensure they render correctly and avoid recursion.

- **Chi Symbol:** The chromatic number is $\chi(G) = 3$.
- **Degrees:** We calculate $\deg(v) = 5$ and $\deg(u) = 2$.
- **Distance:** The distance is $\mathrm{dist}(u, v) = 4$.
- **Set Builder:** Define the neighborhood as $N(v) = \{u \in V \mid uv \in E\}$.

2 2. Theorem System

Theorems should currently be in **Compact Mode** (Name only).

Theorem 2.1 (Hall's Marriage Theorem). (See details)
Theorem `(..) => ..`: Hall's Marriage Theorem

Theorem 2.2 (Euler's Theorem). (See details)
Theorem `(..) => ..`: Euler's Theorem

Switched to Full Mode>

Theorem 2.3 (Bipartite Characterization).

A graph G is bipartite if and only if it contains no odd cycles.

Given: *G is a connected graph*

Proof: ***(⇒) Forward:*** *If G is bipartite, its vertices partition into sets A, B. Any path must alternate between A and B. To return to the start vertex (forming a cycle), we must take an even number of steps ($A \to B \to A$...). Thus, no odd cycles exist.*

(≤) Backward: *Pick a root v. Define levels $L_i = \{u \mid \mathrm{dist}(v, u) = i\}$. If an edge existed between two vertices x, y in the same level L_k, then the path $v \to x$, the edge xy, and the path $y \to v$ would form a cycle of length $k + 1 + k = 2k + 1$ (odd), which is forbidden. Thus, edges only go between levels. We can color even levels A and odd levels B to form a valid bipartition.* □

Theorem `(..) => ..`: Bipartite Characterization

Theorem 2.4 (Whitney's Theorem).

$$\kappa(G) \leq \kappa'(G) \leq \delta(G)$$

To Show: *Vertex Connectivity $\leq$ Edge Connectivity $\leq$ Min Degree*

Proof: ***Right inequality:*** *To isolate a vertex v, we can simply cut all $\deg(v)$ edges connected to it. Thus $\kappa' \leq \delta$.* ***Left inequality:*** *Consider a minimum edge cut of size κ'. If the edges in the cut share a vertex, we can cut that vertex. If not, we can pick one vertex from each edge in the cut to form a vertex cut of size $\leq \kappa'$. (Special care needed for complete graphs, but holds generally).* □

Theorem `(..) => ..`: Whitney's Theorem

3 3. Graph Visualization

We test standard factories and custom adjacency lists.

3.1 Standard Factories

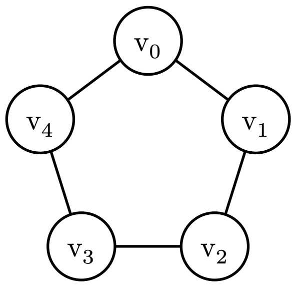


Figure 1: Cycle C_5

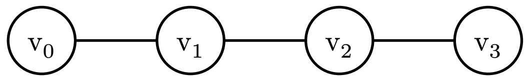
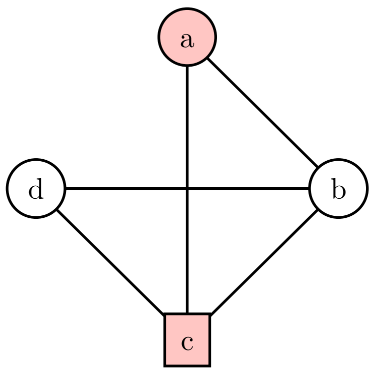


Figure 2: Path P_4

3.2 Smart Highlighting (Algorithm Trace)

We highlight vertices a and c to simulate a path finding algorithm.



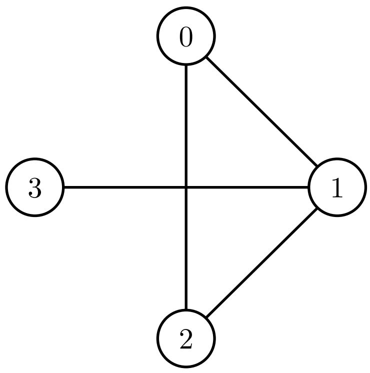
4 4. Matrix Bridge

We define a binary matrix and convert it to a visual graph automatically.

Input Matrix:

$$M = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

Generated Graph:



5 5. Solution Key Toggle

We test the ability to hide/show homework solutions.

Question: What is the size of a perfect matching in $K_{\{2n\}}$?

> *(Solution hidden for study purposes)*

Switched to Solution Mode

> **Solution:**
> Since $K_{\{2n\}}$ has $2n$ vertices, a perfect matching must touch all vertices exactly once. Since each edge consumes 2 vertices, we need $\frac{2n}{2} = n$ edges.