# RamSat: A Ramsey Theory Exploration Suite

Monte Carlo Ramsey Number Estimation, Probabilistic Bounds, and Theoretical Visualizations

## Project Link

**GitHub Project (Click Here)**

## Team


- ***Bhavishy Agrawal [B24CS1023]***
- ***Husain Mohammad Ali [B24CS1084]***
- ***Hayagriv Desikan [B24CS1092]***


## References


1. ***Sr. Prof. Mohit Kumar Jangid, Lecture Slides (Mathematics for Computing), IIT Jodhpur, 2025***
2. F. P. Ramsey, "*On a Problem of Formal Logic*", Proc. London Math. Soc., 1930
3. P. Erdős, "*Some remarks on the theory of graphs*", Bull. AMS, 1959
4. R. L. Graham, B. L. Rothschild, J. H. Spencer, "*Ramsey Theory*", Wiley, 1990
5. V. Chvátal, "*A Note on Ramsey Numbers*", J. Combinatorial Theory, 1970


## Overview

**RamSat** is an interactive and visual computational framework that explores the deep structures of **Ramsey Theory** — the mathematical study of how **order inevitably arises within chaos**.

This project combines **graph theory**, **Monte Carlo simulations**, and **mathematical visualization** to **estimate and demonstrate Ramsey numbers** through computational and theoretical perspectives.

RamSat provides:

- A **Tkinter-based interactive GUI** for simulating Ramsey number estimation.
- **Manim animations** that visually explain key theoretical concepts — *Pigeonhole Principle*, *Erdős Probabilistic Method*, and *SAT-based verification*.
- Real-world insights into how Ramsey theory applies to **protein networks**, **AI graph models**, and **social structures**.

## Core Components

### 1. **Ramsey Number Calculator (Tkinter GUI)**

A fully interactive GUI built with Python's `tkinter`, `networkx`, and `matplotlib`, allowing users to:

- Perform **Monte Carlo simulations** to estimate Ramsey numbers ( R(k_1, k_2) ).
- Visualize **complete graphs with random edge colorings**.
- View **real-time progress**, computation time, and **graph statistics**.
- Switch between **input configuration** and **results view** in a smooth, modern interface.

**Features:**

- ☑ Estimate Ramsey numbers using **probabilistic edge coloring**.
- ☑ Real-time computation feedback and status tracking.
- ☑ Visualization of complete graphs with red-blue color coding.
- ☑ Automatic extraction of graph statistics (vertices, edges, densities).
- ☑ Dark-themed UI with advanced styling for clarity and aesthetics.

---

### 2. **Manim Animation Series**

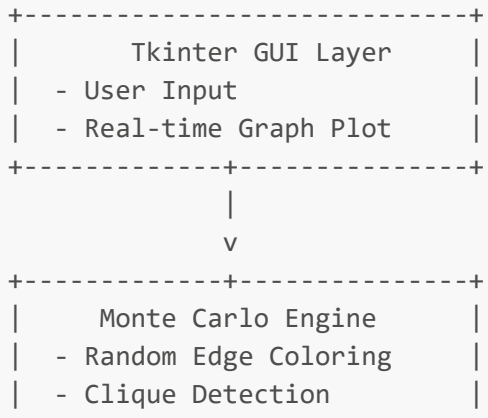
A collection of animated mathematical visualizations built with **Manim CE**, explaining the intuition and mechanics behind Ramsey theory's fundamental methods:

| Animation | Concept Illustrated | Description |
|---|---|---|
| **Erdős Probabilistic Method** | Randomized graph coloring | Shows how random colorings lead to expected Ramsey bounds. |
| **SAT-Based Computational Verification** | Boolean constraint solving | Demonstrates SAT solvers finding guaranteed monochromatic cliques. |
| **Pigeonhole Principle Visualization** | Combinatorial inevitability | Explains why large enough structures always contain order. |
| **Real-World Applications** | Protein, AI, and social networks | Demonstrates structural emergence in biological, neural, and social systems. |

Each animation visually connects abstract combinatorial principles to real computational reasoning and pattern formation.

---

## System Architecture



```
|  - Probabilistic Search     |
+-------------+---------------+
              |
              v
+-------------+---------------+
|   Visualization Layer       |
|  - NetworkX Graphs          |
|  - Matplotlib Canvas        |
+-----------------------------+
```

---

## Theoretical Foundation

> **Ramsey's Theorem:**
> For any positive integers ( r, s ), there exists a minimum number ( R(r, s) ) such that every red–blue coloring of the edges of ( K_{R(r,s)} ) contains either a red ( K_r ) or a blue ( K_s ).

**RamSat** computationally explores this theorem using:

- **Probabilistic simulation** (Erdős method)
- **Monte Carlo clique testing**
- **Combinatorial inevitability (Pigeonhole argument)**
- **SAT formulation for exact verification (conceptual)**

---

## Installation

```
# Clone the repository
git clone https://github.com/yourusername/RamSat.git
cd RamSat

# Create and activate a virtual environment (optional)
python -m venv venv
source venv/bin/activate  # or venv\Scripts\activate on Windows

# Install required dependencies
pip install -r requirements.txt

# Usage

## Run the Ramsey Calculator GUI
python calculator.py
```

## Run Manim Animations

```
# Example: Run Erdős Probabilistic Method animation
manim -pqh animations/erdos_probabilistic_method.py ErdosProbabilisticLowerBound
```

## Example GUI Output

**Input Parameters:**

- Red Clique Size ( k_1 ) = 3
- Blue Clique Size ( k_2 ) = 3
- Trials = 5000
- Max Vertices = 20

**Output:**

```
✓ ESTIMATION COMPLETE
R(3, 3) = 6
Vertices: 6
Edges: 15
Red Edges: 8
Blue Edges: 7
Computation Time: 2.51s
Graph Density: 1.0000
```

**Visualization:**

- Red and blue edges drawn dynamically on a complete graph layout.
- Real-time progress indicator ("Testing n = 15/20 (75%)").

---

## Real-World Applications

| Domain | Description |
|---|---|
| Protein Networks | Detecting consistent structural motifs in protein interaction graphs. |
| Artificial Intelligence | Understanding emergent order in large neural or graph models. |
| Social Networks | Modeling inevitable sub-community formation in large-scale networks. |
| Communication Systems | Predicting failure-resistant connection patterns. |

---

## Tech Stack

| Component | Tool / Library | Purpose |
|---|---|---|
| Core Logic | Python 3.11+ | Primary implementation |
| GUI Framework | Tkinter | Interactive interface |
| Graph Library | NetworkX | Graph construction and analysis |
| Visualization | Matplotlib | Graph plotting in GUI |
| Animation Engine | Manim CE | Mathematical visualization |

| Component | Tool / Library | Purpose |
| --- | --- | --- |
| Parallel Execution | Threading | Background computation |
| Statistics Handling | Dataclasses | Structured storage of graph stats |

## Future Enhancements

◇ SAT-based exact solver integration ◇ Multi-color Ramsey number estimation ◇ GPU-accelerated Monte Carlo simulation ◇ Web-based interactive explorer ◇ AI-based pattern predictor for unknown Ramsey bounds

## License