1 Introduction to the MATH 205 Project

Discrete mathematics is the foundation of fields like computer science, cryptography, and network analysis. In MATH 205, you've explored concepts such as set theory, graph theory, combinatorics, number theory, and finite state machines. The project is your chance to dive deep into one topic, apply these concepts, and showcase your skills through a detailed report.

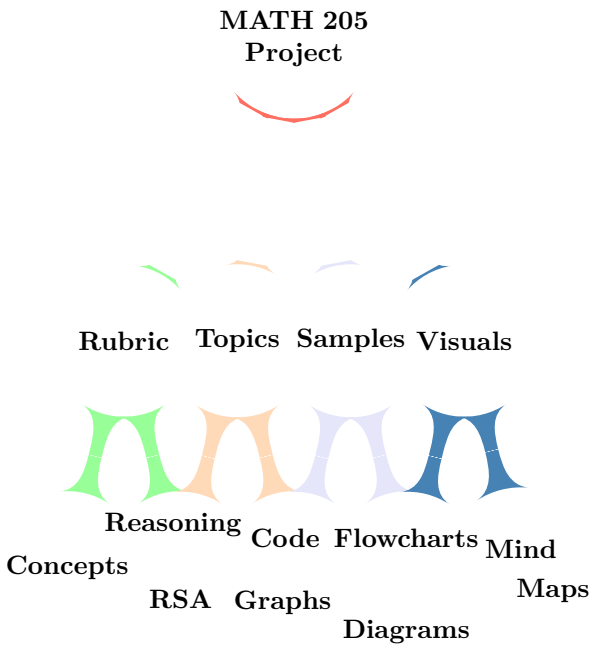

Figure 1: Mind Map of Project Components

المشروع ده زي ما تكون بتبني بيت: لازم تعرف الأساسيات (زي الرياضة المنفصلة اللي درستها)، تختار تصميم حلو (الموضوع اللي هتشتغل عليه)، وتظبط الديكور (التقرير وطريقة العرض).

2 Project Requirements and Rubric

The project is graded out of 10 points, based on a rubric evaluating four categories: Mathematical Concepts, Mathematical Reasoning, Explanation, and Neatness and Organization. The report must have at least three sections: Introduction, Main Body, and Conclusion.

2.1 Project Structure

Your report should include:

- **Problem Description**: Why is the topic important? Include references (e.g., academic papers, textbooks).
- **Introduction**: Connect the topic to MATH 205 concepts.
- **Analysis Tools/Code**: Describe tools (e.g., algorithms) or include code.
- **Further Considerations**: Discuss limitations or extensions.

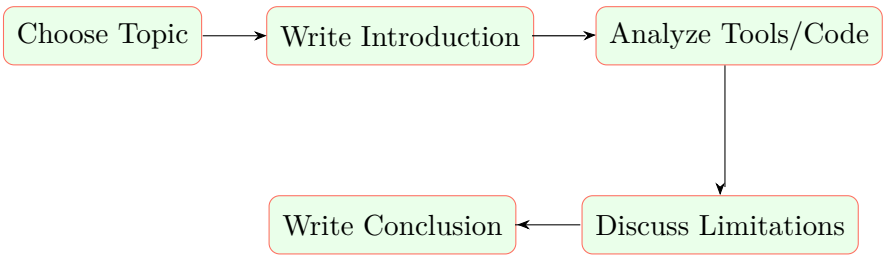

Figure 2: Project Workflow

التقرير بتاعك لازم تبدأ بمقدمة (تعرّف الناس بالموضوع)، وبعدين تدخل في التفاصيل (زي الكود أو الأدوات اللي هتستخدمها)، وفي الآخر تخلّص بملخص حلو وتتكلم عن إيه اللي ممكن يتحسن. الرسمة فوق دي زي خريطة بتفكرك إزاي تمشي خطوة خطوة عشان تبني التقرير.

2.2 Grading Rubric

The rubric is scaled to 10 points from a 100-point distribution. Below is a colorful table, followed by tips and a practice exercise.

Table 1: Project Grading Rubric (Out of 100 Points)

Category	25–20 Points	20–15 Points	15–10 Points	10–5 Points
Mathematical Concepts	Complete understanding; clear definitions and examples.	Substantial understanding; minor gaps.	Some understanding; lacks depth.	Limited or no understanding.
Mathematical Reasoning	Complex, logical reasoning; justified steps.	Effective reasoning; mostly logical.	Some reasoning; inconsistent.	Little or no reasoning.
Explanation	Detailed, clear, and engaging.	Clear but less detailed.	Somewhat unclear; missing components.	Unclear or incomplete.
Neatness and Organization	Professional, easy to read; LaTeX formatted.	Organized; minor formatting issues.	Organized but hard to read.	Sloppy; unorganized.

كل جزء في المشروع بيتحط فيه درجات على حسب إنت عملته إزاي. عشان تجيب الدرجة الكاملة، لازم تفهّم الناس كويس (زي ما تكون بتحكي قصة)، تستخدم منطق قوي (زي لعبة شطرنج)، وتكتب تقرير شكله حلو ومرتب (زي كتاب مصوّر).

2.2.1 Tips for Success

- **Mathematical Concepts**: Define terms clearly (e.g., “A graph is a set of vertices and edges”). Use examples or diagrams.
- **Mathematical Reasoning**: Explain why you chose a method (e.g., “Dijkstra’s algorithm is efficient for sparse graphs”).
- **Explanation**: Use visuals and avoid jargon. Break complex ideas into steps.

3 Project Topic Suggestions

The original description lists 22 topics. We expand on 10, providing context, applications, tools, and visuals to help you choose.

3.1 Expanded Topic List

1. **Finite State Machines (FSMs)**:
   - **Description**: Models systems with finite states, like traffic lights.
   - **Application**: Vending machines, text parsers.
   - **Tools**: Transition diagrams, Python.

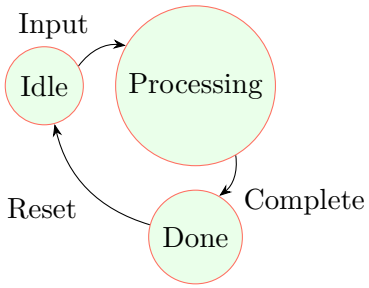

Figure 3: FSM for a Simple Process

2. **RSA and Cryptography**:
   - **Description**: Public-key encryption using number theory.
   - **Application**: Secure online transactions.
   - **Tools**: Modular arithmetic, Python.
3. **Elliptic-Curve Cryptography**:
   - **Description**: Efficient encryption using elliptic curves.
   - **Application**: Blockchain, mobile security.
   - **Tools**: SageMath, curve equations.
4. **Pagerank and Power Method**:
   - **Description**: Ranks web pages using eigenvalues.
   - **Application**: Search engines.
   - **Tools**: NumPy, linear algebra.

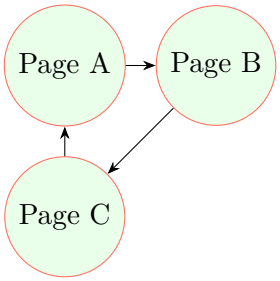

Figure 4: Web Page Link Graph

6. **Markov Chains**:
   - **Description**: Models probabilistic state transitions.
   - **Application**: Weather prediction.
   - **Tools**: Transition matrices, Python.
5. **Compression Algorithms via Binary Trees**:
   - **Description**: Huffman coding for data compression.
   - **Application**: ZIP files.
   - **Tools**: Binary trees, Python.

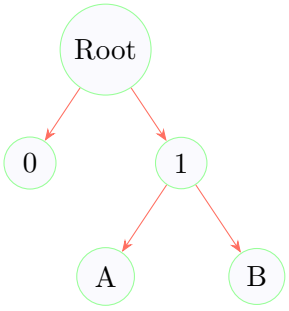

Figure 5: Huffman Tree Example

4 Sample Project 1: RSA and Cryptography

This sample project demonstrates how to structure a report for "RSA and Cryptography," including visuals, code, and tracing.

4.1 Problem Description

RSA secures data using public and private keys, critical for online banking and communication. References: Rivest et al. (1978) and Trappe's "Introduction to Cryptography."

RSA ده زي إنك تبعت رسالة في صندوق مقفول، محدش يعرف يفتحه غير اللي عنده المفتاح الخاص. الموضوع ده مهم عشان بنستخدمه في حاجات زي الدفع أونلاين أو إيميلات سرية.

4.2 Introduction

MATH 205 covers number theory, which RSA uses to ensure security through prime factorization.

في الكورس، اتعلمنا عن الأعداد الأولية والحسابات اللي بنستخدمها عشان نحمي البيانات. RSA بيستخدم الأفكار دي زي لو عندك لغز رياضي صعب محدش يعرف يحله غيرك.

4.3 Main Body

4.3.1 Mathematical Concepts

RSA uses:

- Prime numbers: p, q.
- Modulus: $n = p \cdot q$.
- Totient: $\phi(n) = (p-1)(q-1)$.
- Keys: Public (e, n), private (d, n).

RSA Key Generation Example

For $p = 61$, $q = 53$:

- $n = 61 \cdot 53 = 3233$.
- $\phi(n) = 60 \cdot 52 = 3120$.
- Choose $e = 17$.
- Compute $d = 2753$ (using extended Euclidean algorithm).

RSA زي لعبة أرقام: تختار رقمين أوليين كبار (زي 61 و53)، تضربهم في بعض، وبعدين تستخدم حسابات