Honors Discrete Mathematics: Course Project Topics

Professor Gerandy Brito Spring 2023

Each topic is given its own page. Loosely, Sarthak will be mentoring projects 1-10, and Niti will be mentoring projects 11-20. Professor Brito will serve as a mentor for both projects.

Niti Jayakumar, Sarthak Mohanty

1   Genetic Algorithms

Genetic Algorithms (GAs) are a type of optimization algorithm inspired by the process of natural selection and genetics. They are used to find an optimal solution to a problem by mimicking the process of evolution.

A general framework for a GA starts with a set of solutions called the population. Each solution is represented by a chromosome, which is a binary string of bits. The chromosomes are then evolved through several generations, with each generation having a better solution than the previous one. The evolution process is done using a combination of selection, crossover, and mutation operations.

Selection is the process of choosing the best solutions from the current population to form the next generation. This is done using a fitness function that evaluates each solution and assigns it a score. The solutions with the highest scores are chosen as parents to create offspring for the next generation.

Crossover is the process of combining two solutions to create a new solution. It is done by randomly selecting a crossover point in the binary string of each parent and then swapping the bits between the two parents to create the offspring. This process creates new solutions that have a combination of the best traits from both parents.

Mutation is the process of randomly changing bits in the binary string to introduce new solutions into the population. This helps maintain genetic diversity and allows the GA to explore new solutions.

Main ideas for this project can go in many directions. You could explore the learning aspect through deep neural networks. You could also analyze the implementation of GAs and how they use techniques such as particle swarm or simulated annealing. A couple of possible areas for extension or generalizations are as follows:

- Study of hybrid genetic algorithms that combine GAs with other optimization algorithms

- Study of parallel and distributed GAs, including island models and particle swarm optimization

- Application of GAs to real-world problems, including optimization of engineering designs and financial portfolios

2   Cryptography in the Modern Age

Cryptography plays a vital role in ensuring secure communication in the modern age. In this project, we will be focusing on modern encryption algorithms, specifically Advanced Encryption Standard (AES). AES is a symmetric-key encryption algorithm that is widely used in various applications such as internet security protocols, electronic funds transfers, and secure communications.

The implementation of an encryption system using AES will be one of the main deliverables in this project. In addition to the implementation, we will delve into the mathematical foundations of AES and other encryption algorithms. This will include an in-depth analysis of the underlying mathematical concepts such as number theory, Galois fields, and cryptography-specific concepts like trapdoor functions.

A trapdoor function is a mathematical function that is easy to compute in one direction but difficult to compute in the opposite direction without knowing some additional information, called the trapdoor.

As an example, the trapdoor function in RSA encryption is the modular exponentiation function, $c = m^e$ mod N, where m is the plaintext, e is the public exponent, N is the modulus and c is the ciphertext. The trapdoor information is the private key, which consists of the factors of N. In the opposite direction, decryption is performed by computing $m = c^d$ mod N, where d is the private exponent derived from the private key.

Another example is the trapdoor function in Elliptic Curve Cryptography (ECC). In ECC, the trapdoor function is the scalar multiplication operation $Q = kP$, where P is a point on the elliptic curve, k is the scalar and Q is the result of the scalar multiplication. The trapdoor information is the private key, which is the scalar k. Decryption is performed by computing $Q = -kP$.

The trapdoor function in AES encryption is based on the substitution-permutation network (SPN) structure. The SPN structure consists of repeated rounds of substitution and permutation operations, where the substitution operation is performed by the use of S-boxes, and the permutation operation is performed by the use of linear transformations. The S-boxes are non-linear functions that scramble the input data, and their inverse operations are difficult to compute without the knowledge of the key used in AES encryption.

Specifically, let x be the plaintext and y be the ciphertext. The encryption operation in AES can be expressed as $y = E(x, K)$, where K is the secret key. The S-boxes used in AES are non-linear functions $S(x)$, and their implementation in AES can be expressed as:

$$y = S(x) \oplus K,$$

where $\oplus$ represents the bitwise exclusive-or operation. The inverse operation, decryption, can be expressed as:

$$x = S^{-1}(y) \oplus K$$

The inverse operation $S^{-1}(y)$ is difficult to compute without knowledge of the key K. This is the essence of the trapdoor function in AES encryption. The implementation of the S-boxes and their properties, as well as the use of the key schedule, make AES a secure encryption algorithm.

I recommend using this comic as a starting point.

3   Recursive Algorithms and Fractals

A *fractal* is a self-similar geometric shape that can be divided into smaller copies of itself. One way to generate fractals is through the use of recursive algorithms.

One famous example of a fractal is the *Koch snowflake*, which is created by starting with an equilateral triangle, and then dividing each line segment into three equal parts, and replacing the middle part with two line segments that form a smaller equilateral triangle. This process is repeated an infinite number of times, creating a fractal shape with an infinitely complex boundary. The Koch snowflake can be described mathematically as a fractal curve with the fractal dimension, D, which is a measure of its roughness, given by the equation $D = \log_3 4$.

Another example of a fractal is the *Mandelbrot set*, which is a set of complex numbers that exhibit complex, self-similar patterns when plotted on a graph. The algorithm for generating the Mandelbrot set is also recursive, and involves iterating the function $f_c(z) = z^2 + c$ for each value of c in the complex plane. The boundary of the Mandelbrot set is a fractal shape with a fractal dimension given by $D = 2$.

The mathematics involved in fractals and recursive algorithms includes complex numbers, iterative algorithms, and infinite sequences. The complex numbers are represented in the form $z = a + bi$, where a and b are real numbers and i is the imaginary unit, $i^2 = -1$. The iterative algorithm for the Mandelbrot set involves checking the behavior of the sequence defined by the recursive formula $z_{n+1} = f_c(z_n) = z_n^2 + c$ starting from $z_0 = 0$. The sequence is considered to be a member of the Mandelbrot set if it remains bounded in the complex plane, i.e. if $|z_n| < \infty$ for all n.

Your main task for this project should be where you demonstrate some of the recursive techniques that you have learned to create interesting fractals. It should also include some coding implementation of your techniques.

A couple of possible areas for extension or generalization are as follows:

- Study different types of fractals and the algorithms used to generate them, including the *Sierpinski triangle*, *dragon curve*, and *Julia sets*.

- Explore the properties of fractals, including their dimensionality and self-similarity, and how these properties relate to the underlying mathematical concepts, such as complex dynamics and iterated function systems.

- Study applications of fractals and recursive algorithms in areas such as computer graphics, data compression, and chaos theory, where the self-similarity and infinite detail of fractals can be utilized to achieve efficient representations of data and to model complex, dynamic systems.

4　Lambda Calculus

Lambda calculus is a mathematical formalism that is used to describe and study computation and algorithms. It was first introduced by Alonzo Church in the 1930s, and has since become an important area of research in theoretical computer science, programming languages, and logic.

In lambda calculus, functions are treated as first-class citizens and can be passed as arguments to other functions, assigned to variables, and manipulated in other ways. The basic building block of lambda calculus is the *lambda expression*, which is used to define anonymous functions. A lambda expression has the form

$$\lambda x.\ M$$

where x is the argument of the function and M is the expression representing the function's body.

The evaluation of a lambda expression is performed by substituting the argument x in M with a specific value and reducing the expression until it can no longer be simplified. This process is called *beta reduction*. The rules of beta reduction are well-defined and allow for the calculation of the value of a lambda expression.

Here are a few examples of lambda expressions:

- $\lambda x.\ x$ is the identity function, which simply returns its argument.

- $\lambda x.\ x + 1$ is a function that adds 1 to its argument.

- $\lambda x.\ \lambda y.\ x + y$ is a function that takes two arguments and returns their sum.

Lambda calculus has many interesting properties and features, including the ability to represent any computable function, the capability to encode simple data structures and algorithms, and the property of Turing completeness, which means that it is capable of simulating any algorithm that can be implemented on a Turing machine.

The aim of this project is to study the theory of lambda calculus, including its syntax, semantics, and reduction rules, as well as its applications in programming languages and computer science. In addition, students will implement a simple interpreter for lambda calculus in order to gain a deeper understanding of the subject.

Possible areas for extension or generalizations are as follows:

- Consider the theory of combinatory logic, which is a close relative of lambda calculus. Combinatory logic is characterized by the absence of variables, and as such, all functions are anonymous. The goal of this extension would be to study the theory of combinatory logic and to compare it with lambda calculus.

- Study the theory of type systems and type inference in the context of lambda calculus. This would include an introduction to Hindley-Milner type systems, the study of type inference algorithms, and the relationship between type systems and the ability to reason about the behavior of programs.

- Consider the use of lambda calculus in functional programming languages, such as Haskell, Scheme, and ML. Study the theory and practice of functional programming, and implement simple programs in one or more functional programming languages.

5   Godel's Incompleteness Theorem

Godel's Incompleteness Theorem is one of the most famous results in the foundations of mathematics. It was proved by Kurt Gödel in the early 20th century and states that, within any sufficiently powerful mathematical system, there will always exist statements that are true, but unprovable within the system.

To understand the theorem, we first need to understand the idea of Godel numbering. Given a mathematical system, we can assign a unique integer (referred to as a "Godel number") to each mathematical statement, symbol, or proof. This allows us to encode mathematical statements as integers, and perform operations on them using arithmetic. For example, the statement "$2 + 2 = 4$" can be encoded as the integer 123456. The statement "For all x, the statement '$x + x = 2x$' is true" can also be encoded as an integer, say 654321.

Once we have assigned Godel numbers to our mathematical statements, we can use these numbers to reason about the statements themselves. For example, we can define the statement "A implies B" as the statement "$\neg A \lor B$", where $\neg$ represents negation and $\lor$ represents disjunction. This means that if A is the statement "$x + x = 2x$" and B is the statement "$x = x$", then the statement "A implies B" is equivalent to the statement "$\neg(x + x = 2x) \lor (x = x)$". We can even encode this statement as its own Godel number! This encoding is how Godel proved his Incompleteness Theorems.

Godel's Incompleteness Theorem is significant because it shows that within any sufficiently powerful mathematical system, there will always be statements that are true but cannot be proven within the system itself. This highlights the limits of mathematical systems and the need for a more comprehensive understanding of mathematical truth. The theorem is a crucial contribution to the foundations of mathematics and continues to be a topic of interest and research in the field today. In this project, you'll have the opportunity to further understand the theorem by encoding mathematical statements as Godel numbers and studying the first and second incompleteness theorems in detail.

I **highly** recommend using this article as a starting point.

6   Combinatorial Games

A combinatorial game is a two-player game with perfect information, where the players make moves in turn, with the aim of being the last player to make a move. To illustrate, in this overview we focus on two particular games: Sprouts, and Dots and Boxes.

Sprouts: Sprouts is a game played on a sheet of paper, with a pen. The game starts with a dot on the paper, and players take turns adding a new dot and connecting it to one of the existing dots by a line. The line cannot cross over any other lines. The game ends when a player is unable to add a new dot. The player who adds the last dot is the winner.

Dots and Boxes: Dots and Boxes is played on a square grid of dots. The players take turns connecting two adjacent dots by a line. If a player completes a box, they write their symbol inside the box and take another turn. The game ends when all boxes have been completed. The player with the most boxes is the winner.

To study these games, we will use the theory of impartial combinatorial games, which assigns a mathematical value to each game, known as the nim-value or Grundy number. This allows us to determine who has a winning strategy and who does not, and also to combine games to form more complex games.

Sprague-Grundy Theorem. Let $G_1, G_2, \ldots, G_n$ be combinatorial games. The game $G = \bigoplus_{i=1}^{n} G_i$ is defined to be the game where players can make moves in any of the games $G_1, G_2, \ldots, G_n$ in any order they choose, but once a player is unable to move in a game, they cannot make any further moves in that game. Then the nim-value of the game G is equal to the nim-value of the game $G_1 \oplus G_2 \oplus \cdots \oplus G_n$.

A couple of possible areas for extension or generalizations are as follows:

- Consider the game of NIM, where the players take turns removing stones from a heap. Investigate the relationship between the size of the heap and the nim-value of the game.

- Generalize the Sprague-Grundy theorem to m-player games, where $m \geq 2$ players take turns making moves. Investigate how to calculate the nim-value of such games, and the conditions under which a player has a winning strategy.

- Consider games where the moves are not deterministic. Investigate how to extend the Sprague-Grundy theorem to these games, and the conditions under which a player has a winning strategy.

7 Create Your Own Game

In this project, you will design, create, and analyze a math-based game. The game must be complex enough to provide a challenge to players, and must involve a significant mathematical component, such as strategy and problem-solving.

There are many different types of math-based games to choose from, but yours should be completely unique. When selecting a game to create, consider the following questions:

- What mathematical concepts are involved in the game?

- What is the objective of the game?

- How is the game played? What are the rules and how do they impact strategy?

- What are some examples of similar games and what makes your game unique?

Once you have selected a game to create, you will need to carefully design and implement it. This will involve creating rules, designing a board or playing field, creating pieces or tokens, and possibly even coding a computer version of the game.

After you have created the game, you will need to analyze it. This will involve studying the various strategies and tactics that players can use, as well as analyzing the mathematical aspects of the game. You may also want to play the game yourself or have others play it to see how it works in practice.

The final step of the project will be to write a report on your game. This should include the following elements:

- An introduction to the game and its mathematical foundations

- A description of the rules and how the game is played

- An analysis of the game's mathematical properties, such as strategies, tactics, and any relevant theorems or equations

- An explanation of any coding or programming that was involved in creating the game

- A discussion of any challenges or limitations encountered during the creation process

- Conclusion and future directions for the game.

Good luck and have fun creating your math-based game!

8   Monte Carlo Methods

Monte Carlo methods are a family of computational algorithms that rely on repeated random sampling to compute numerical results. They are widely used in fields such as physics, finance, and engineering, among others, where mathematical models are too complex to solve analytically.

The basic idea behind Monte Carlo methods is to generate random samples from a given distribution, and then use these samples to estimate the desired quantity. For example, in physics, Monte Carlo methods can be used to compute the properties of complex systems, such as molecules or materials, by generating random configurations and computing the properties of interest for each one.

Your main idea can be whatever you desire, but in this overview we will focus on a specific type of Monte Carlo method: Markov Chain Monte Carlo (MCMC). MCMC methods are a class of algorithms that generate random samples from a target distribution by constructing a Markov Chain that has the target distribution as its stationary distribution. The samples generated by the MCMC method are then used to estimate the desired quantity.

The main theorem in this area is the Metropolis-Hastings algorithm, which provides a general framework for constructing MCMC methods. The Metropolis-Hastings algorithm works as follows:

- Choose an arbitrary starting state.

- At each step, generate a proposal state from the current state using a proposal distribution.

- Compute the acceptance probability of the proposal state, based on the target distribution and the proposal distribution.

- With probability equal to the acceptance probability, accept the proposal state as the next state. Otherwise, reject the proposal state and stay in the current state.

- Repeat the above steps to generate a sequence of states, which form a Markov Chain.

The Metropolis-Hastings algorithm can be seen as a generalization of the Metropolis algorithm, which is a special case of the Metropolis-Hastings algorithm where the proposal distribution is symmetric.

A couple of possible areas for extension or generalizations are as follows:

- Consider different types of proposal distributions, such as Gaussian distributions or random walks, and compare their performance in terms of convergence to the target distribution and computational efficiency.

- Consider the use of parallel and distributed computing techniques to speed up the computation. For example, consider using GPU programming or cloud computing to run large-scale Monte Carlo simulations.

- Consider the use of adaptive MCMC methods, which dynamically update the proposal distribution based on the history of the Markov Chain, in order to improve the convergence and efficiency of the simulation.

9   Association Rule Learning

Association Rule Learning is a data mining technique used to uncover the relationships between items within large datasets. Given a set of items $I = i_1, i_2, ..., i_m$ and a set of transactions $T = t_1, t_2, ..., t_n$, where each transaction is a set of items, the goal of association rule learning is to find rules of the form $X \to Y$, where X and Y are subsets of I, that are considered to be strongly associated within the transactions.

One of the most common measures of association strength is support, defined as the fraction of transactions that contain both X and Y:

$$\text{Support}(X \to Y) = \frac{|t \in T \mid X \subseteq t \text{ and } Y \subseteq t|}{|T|}$$

Another common measure of association strength is confidence, defined as the fraction of transactions that contain X that also contain Y:

$$\text{Confidence}(X \to Y) = \frac{|t \in T \mid X \subseteq t \text{ and } Y \subseteq t|}{|t \in T \mid X \subseteq t|}$$

A common approach to finding association rules is to set a minimum threshold for both support and confidence and then use an algorithm to identify all the rules that meet these requirements. A well-known algorithm for this is the Apriori algorithm, which uses the concept of itemset lattices to reduce the number of item combinations that need to be considered.

There are several areas for extension or generalization of association rule learning, including:

- Using alternative measures of association strength, such as lift, conviction, or interestingness.

- Using algorithms that incorporate domain knowledge, such as decision trees, clustering, or Bayesian networks.

- Incorporating additional information, such as transaction timestamps, demographic data, or contextual data.

- Handling missing or noisy data, such as using imputation techniques or data cleaning methods.

- Scalability, such as implementing parallel or distributed algorithms to handle extremely large datasets.

This project may involve exploring one or more of these areas, conducting a comprehensive comparison of different algorithms, or applying association rule learning to a real-world problem and evaluating its results.

10   Chaos Theory and Iterated Function Systems

Chaos theory is a branch of mathematics that studies the behavior of systems that are highly sensitive to initial conditions. This means that small differences in the starting conditions of a system can lead to vastly different outcomes over time. The most well-known example of this phenomenon is the butterfly effect, where the flapping of a butterfly's wings in one part of the world can cause a chain of events that leads to a tornado in another part of the world.

In mathematics, chaos theory is often studied using iterated function systems (IFS). An IFS is a collection of functions $f_1, f_2, ..., f_n$ that are applied to a point in a metric space. The functions can be contractive, meaning that they reduce the distances between points, or they can be expanding, meaning that they increase the distances between points. In either case, the behavior of the system is highly dependent on the choice of functions and the starting point.

IFS can be used to generate fractal images, which are images with self-similar patterns at different scales. The fractal images generated by IFS are often visually striking and can be used to model a wide range of natural phenomena, such as coastlines, mountains, and galaxies.

The aim of this project is to study the theory of chaos and iterated function systems, including the mathematical foundations, algorithms for generating fractals, and applications to real-world problems. Students will implement code to generate fractals and will explore the behavior of IFS with different choices of functions and starting points.

Possible areas for extension or generalization are as follows:

- Study the relationship between IFS and attractors, which are sets that the iterates of a system tend to converge to. Investigate the different types of attractors, including fixed points, periodic points, and chaotic attractors.

- Consider the use of IFS in image compression. Study the theory and practice of image compression, and implement simple algorithms for compressing and decompressing images using IFS.

- Study the theory of cellular automata, which are discrete dynamical systems that can be used to model complex behavior. Investigate the relationship between cellular automata and IFS, and implement simple cellular automata models.

11   The real numbers as the (topological) closure of the rationals.

Explore the notion of *field of numbers* and their applications.

The rationals are defined as $\mathbb{Q} = \{x/y : x, y \in \mathbb{Z}\}$, or the set of all fractions of integers. This contains numbers like $1/2$, $-52/3$, and 100, but not numbers like $\sqrt{2}$ or π or $0.110100100010000\ldots$ In order to reach those numbers, we can take one main approach to closure: a sequence of rational numbers (a_i) corresponds to a unique real number if it has a limit.

How do we define a sequence having a limit? You have probably heard of epsilon-delta proofs for the existence of limits. In our case, we will use an epsilon-N proof to define the sequence of rational numbers (a_i) approaching a limit of a real number r. The following statement is the statement we need to use to prove that such a sequence approaches a limit:

$$\forall \epsilon \exists N \forall n \in \mathbb{Z}((\epsilon > 0 \wedge n > N) \rightarrow |a_n - r| < \epsilon).$$

We assert that a real number exists if there is some sequence approaching it. For most real numbers that we're interested in, this is pretty easy because we have a decimal expansion. For example, we associated $\sqrt{2}$ with $\{1, 1.4, 1.41, 1.414, 1.4142, 1.41421, 1.414213, 1.4142135, \ldots\}$, so this number exists!

This is called a topological closure because topology is the tool that we use to handle points being arbitrarily close to each other without actually touching.

The real numbers are in a category of objects known as fields: objects with a notion of commutative addition, commutative multiplication (related by the distributive property $a \cdot (b + c) = a \cdot b + a \cdot c$), a 0 element, a 1 element, and division and subtraction. The complex numbers and the rationals are also fields, but there are also some more exotic objects: the finite fields $\mathbb{F}_p$ only have a finite number of elements, but we can use all of the field theorems we've learned on them!

You will investigate these and other related concepts in the field of "Abstract Algebra."

Ideas for directions to go in:

- Abel-Ruffini Theorem: there is no solution to radicals to general polynomial equations of degree five or higher with arbitrary coefficients.

- Cayley's Theorem: every group is isomorphic to the group of permutations of n objects for some number.

- Gauss's Theorem (primitivity): If P(x) and Q(x) are primitive polynomials over the integers, their product P(x)Q(x) is also primitive.

- In the same way that vector spaces are defined over fields, modules are defined over rings. Investigate the ways that the properties of vector spaces like the Basis Theorem might extend to modules.

12   What does it mean to be irrational? Proofs and extensions.

The rational numbers are a fairly well-behaved set! We can easily write out any of its elements as a fraction of two integers, and we expect them to act fairly across the whole set with every rational number being accessible from the others by operations of division, addition, subtraction, multiplication, etc. However, the real numbers are not so nicely behaved. $\sqrt{2}$ is irrational, meaning that it is not accessible from the rational numbers by such operations. However, we might think of the number $2\sqrt{2}$ or $\sqrt{2}/2 + 1/50$ as being "co-rational" with it. This is known as "commensurable," and commensurable numbers can be infinitely well approximated by each other. That is, using the rational numbers we can find equalities in the same field. We're also interested in when we can't necessarily approximate something arbitrarily well. The number ϕ, the golden ratio, is very unrational, in that it is very hard to find a good approximation of it (the denominator has to be far larger in order to find good approximations). Hurwitz's Irrational Number theorem provides bounds of how good these approximations are, until we allow multiplying by the irrational ϕ, in which case the bound gets better. Interestingly, there is an infinite collection of numbers like this that can't be well-approximated, but the bound continues to fall in accordance with a series called the Lagrange Numbers. You can investigate the properties of this series and the theorem.

There are also other types of irrationality, for example algebraic irrationality, where a number isn't the solution to a polynomial equation.

Some ideas:

- Gelfond-Schneider Theorem: If a and b are algebraic numbers with $a \neq 0$ and 1 and b are irrational, then any value of a^b is a transcendental number.

- Approximation of algebraic numbers: Thue-Siegel-Roth theorem.

- Lonely Runner Conjecture.

13   The Odds are in Your Favor: Exploring Discrete Probability

A discrete setting is a mathematical model where data takes on a countable number of possible values, such as integers, edges, or paths. Discrete probability is the branch of mathematics that deals with the study of discrete random variables and their distributions. In this project, we will explore the relationship between discrete probability and counting. We will examine several classic examples, including Catalan numbers, Dyck's paths, and the Semicircle Law.

Catalan numbers, C_n, are a sequence of integers that appear in various counting problems. They are often used to count the number of ways to construct objects with certain restrictions. For example, consider the problem of counting the number of ways to parenthesize n pairs of parentheses so that the expression is well-formed. The solution is C_n. The formula for Catalan numbers is given by the following equation:

$$C_n = \frac{1}{n+1} \binom{2n}{n}$$

Another way to define Catalan numbers is as follows: how many ways can you traverse an $n \times n$ grid of lattice points to the top right, only moving up and to the right without crossing over the diagonal? This corresponds to the number of *Dyck paths* of order n.

Now consider a similar problem: taking a $n \times n$ grid of lattice points, starting from the bottom left corner and going to the top right, how many paths can you take *including moves going down and left as well as right and up*? How might you estimate how many of these paths are there? You can take a random sample of them by generating them fairly, and compute the probability of each given path and then take the reciprocal of that. Because we can expect, if a path has probability 10^{-22}, then there are 10^{22} similar paths.

Finally, we move on to one more interesting result, the Semicircle Law, which states that the limiting distribution of the eigenvalues of a random matrix is given by the following equation:

$$f(x) = \frac{1}{2\pi} \sqrt{4 - x^2}$$

for $-2 \leq x \leq 2$ and zero otherwise. Note how this looks like the formula of a semicircle. Why? Mathematicians have found a theorem of this statement that was first empirically verified, and one you should explore in depth.

In the course project, students will consider the following possible areas for extension or generalizations:

- Analyze the relationship between the generating function of Catalan numbers and the generating function of Dyck's paths.

- Investigate the distribution of the number of times a random Dyck path returns to the x-axis.

- Consider the case where the steps in a random walk are not symmetric and derive the limiting distribution in this case.

- Develop a method for generating random samples from the Semicircle Law and evaluate its performance.

14   Set: The Pattern-Matching Card Game

Set is a pattern-matching card game where the goal is to identify a "set" of three cards from a deck of twelve cards. Each card has four attributes: color, shape, number, and shading. Each attribute can take one of three values, so there are a total of $3^4 = 81$ possible cards in a deck.

In 2016, with the help of a combinatorics professor right here at Georgia Tech, a major breakthrough in the understanding of the mathematical foundations of the game of Set was made, as described in the article here. Try to understand what progress was made, and what techniques were used.

The study of Set and its mathematical foundations is ongoing, with recent developments exploring the application of group theory and graph theory to the game. In particular, there has been interest in understanding the number of sets in a deck of n cards, for larger values of n. A natural extension of the previous result would be to determine the maximum number of sets in a deck of n cards, as a function of n.

Another area of ongoing research in the mathematical study of Set is the study of strategies for playing the game. Here, one can consider various approaches, including combinatorial and algorithmic approaches, and analyze their efficiency and effectiveness in finding sets. A central question in this area is to determine the minimum number of cards that need to be revealed in order to guarantee the discovery of a set, or to determine the expected number of cards that need to be revealed, given a random distribution of cards.

There are several possible areas for extension or generalization in the mathematical study of Set:

- Consider the case where there are additional rules or constraints on the game, such as restrictions on the order in which cards can be selected, or restrictions on the number of sets that can be selected in a given turn. Analyze the impact of these changes on the game, and determine the new maximum number of sets and the new strategies for finding sets in the deck.

- Analyze the corresponding n-category versions of the problems, and especially try to determine the probability distributions occurring in this case.

15   Patterns in Primes

Prime numbers are an important topic in mathematics, with a rich history and numerous famous theorems and conjectures associated with them.

- Legendre's conjecture: Between the number n^2 and $(n+1)^2$ for natural number n, prove there is a prime.

- Bertram's postulate: prove that there is a prime number in between n and $2n$.

- The prime number theorem: the number of primes below n is proportionate to $\frac{n}{\log n}$.

A couple of possible areas for extension or generalizations are as follows:

- Generalized Prime Number Theorem: Study the distribution of prime numbers in various number systems, such as algebraic integers and p-adic integers.

- Prime Number Race: Study the distribution of prime numbers in different sequences and find the fastest growing sequence of primes.

- Analytic Number Theory: Apply complex analysis and other advanced mathematical tools to study the distribution of prime numbers.

16 Sylow's Theorems and the complete categorization of all finite simple groups

Lagrange's theorem is an important theorem in group theory for a finite group G and its subgroup H. The order of H divides the order of G, however in order to discover the structure of finite groups in general, Langrange's theorem is not enough, because it's not guaranteed that such a subgroup exists. Sylow's Theorems solve this problem. Every group is guaranteed to have a Sylow p-group, where p is a prime, of the size p^n, where p^n is the size of the largest prime power that divides the order of G. This can prove cool theorems like all groups with 15 elements are commutative (abelian).

But there are even more implications of this theorem, including extremely important work on the factoring of all finite groups (which led us to understanding the special status of the sporadic groups!).

When you're investigating this, talk about the applications of the factorization of groups, the sporadic groups (and the difference between the Happy Family and the 6 outliers). Also discuss Lie groups, the methods used to find these groups, and how Sylow's theorems helped this process.

17   Integer Partitions

Let n be a positive integer. A partition of n, also called an integer partition, is a way of writing n as a sum of several positive integers. For example, 3 can be partitioned in 3 ways: $3, 2 + 1$, and $1 + 1 + 1$. Consider the function $p(n)$, representing the number of possible patitions of a non-negative integer n. E.g. $p(3) = 3$. Things to consider

- Think about finding an expression for p. Start with the generating function.

- Is the number of self-conjugate partitions the same as the number of partitions with distinct odd parts? Self-conjugate partitions are partitions that are conjugates itself, where a conjugate of a partition is derived from the representation of the partition using Ferrers' Diagramming method. When you flip the partition along the diagonal, you get the conjugate of the partition.

- Ramanujan's congruences consist of the following modular congruences:

$$p(5k + 4) \equiv 0 \pmod{5},$$

$$p(7k + 5) \equiv 0 \pmod{7},$$

$$p(11k + 6) \equiv 0 \pmod{11}.$$

How are they related? Explore these congruences.

18   Carmichael Numbers

(Quanta Magazine). How does one test if a number is prime? In contrast to the foolproof methods we have (like checking if every single number less than n divides n), we can explore the Fermat Primality Test. This is a way to test if a number is prime using Fermat's Little Theorem. However, it fails an infinite number of times! Certain numbers, known as the Carmichael numbers, pass this test even though they are not prime. Explore these numbers (a good starting point is the research discussed in the Quanta Magazine article).

19　Four Color Theorem

(**) Graph theory is essential to computer science, networks theory, and the topology of closed connected regions. In 2-dimensions, closed connected regions tile the plane, and this can be seen in the context of a map containing countries with no exclaves. Mathematicians were able to show the seven-color theorem—that any graph could be colored with seven colors on each country, proven by mapping it to a "graph": nodes representing countries, and edges between them representing borders. Theorems in this reduced-complexity system (not having to deal with the detail of coastlines, and being equivalent in all respects) permits the concept of planar graphs.

20   Famous Unsolved Problems in Mathematics

[nithya I combined your collatz conjecture and riemann hypothesis topics into this one - sarthak]

A *mathematical problem* is a statement or question that can be answered through mathematical methods and reasoning. Many of these problems have been solved over time, however, some remain unsolved.

In this project, we will focus on some of the most famous and important unsolved problems in mathematics. These problems come from a variety of areas such as number theory, geometry, analysis, and combinatorics.

One of the oldest unsolved problems is the *Pell's equation*, which is a Diophantine equation of the form

$$x^2 - dy^2 = 1$$

where $d \in \mathbb{N}$ and d is not a perfect square. The aim of this project is to find an algorithm that finds all solutions to Pell's equation, or prove that no such algorithm exists.

Another famous problem is the *Riemann Hypothesis*, which states that the non-trivial zeros of the Riemann zeta function $\zeta(s)$ are all located on the critical line of $1/2$. Proving or disproving this hypothesis remains one of the most important open problems in mathematics, and it has significant implications in number theory and prime number distribution.

Another intriguing problem is the *Collatz Conjecture*, which is a conjecture in the field of discrete mathematics. Given any positive integer n, we define the following sequence:

$$n \to \frac{n}{2} \quad \text{if } n \text{ is even,}$$

$$n \to 3n + 1 \quad \text{if } n \text{ is odd.}$$

The Collatz Conjecture states that this sequence always reaches 1, regardless of the starting value of n.

Your main task will be to take a close look at some of these problems, read research papers, and try to udnrestand some of the recent developments surrounding the problem. **In my opinion, this is the most difficult project.**

A couple of possible areas for extension or generalizations are as follows:

- Consider other famous unsolved problems such as the *Goldbach Conjecture* (every even integer greater than 2 can be expressed as the sum of two prime numbers), the *Hodge Conjecture* (compact Kähler manifolds are projective), or the *Birch and Swinnerton-Dyer Conjecture* (the rank of an elliptic curve is equal to the order of vanishing at infinity of its L-function). Research these problems, and try to create visualizations.

- Consider a problem from another area of mathematics, such as *Navier-Stokes Equation* (describing the motion of fluids), *Yang-Mills Equation* (describing the behavior of particles in quantum field theory), or the *Mass Gap Conjecture* (massless particles cannot appear in theories of massive particles). Research these problems and attempt to find some significant results.