CALIFORNIA STATE SCIENCE FAIR
2017 PROJECT SUMMARY

Name(s) Daniel L. Bishop	**Project Number** J1501

Project Title

The Mathematics of the Mandelbrot and Julia Sets

Abstract

Objectives/Goals

My goal is to understand where the Julia and Mandelbrot sets come from and answer many intriguing questions: How many different types of Julia sets are there? Do Julia sets of complex conjugate c values always have the same symmetry (like reflection across the imaginary axis) and if so why? Are the causes of the infinite repetition such as those in swirling patterns understood? Can they seem obvious if you understand the math well enough? If the critical orbit (the orbit of z = 0) escapes, is the hole left repeated "fractally" everywhere so there is no area that doesn#t have holes or gaps? Is this the origin of the fractal dust Julia Sets? Are these just like the cantor set? Does the Mandelbrot set contain every Julia set? How is the Mandelbrot set related to the Julia sets? Is every point on the Mandelbrot set the center point for a Julia set with the same parameters?

Methods/Materials

I studied the mathematics behind these sets: trigonometry, complex numbers, and complex number geometry. I used the Java programming language to visualize complex behavior of iterations and search for patterns in fractals.

Results

I was able to answer many questions I had about the Mandelbrot and Julia sets, but many remain. I generated my own Julia sets for different c values and then expanded on interesting areas where the images changed dramatically for small changes in c. From these I was able to see symmetries: for example that complex conjugate c values are indeed mirror images across the imaginary axis. I found that if the center of a Julia set escapes, then it is a Cantor Set; every point eventually escapes and no region is free of holes, no matter how small.

Conclusions/Discussion

When I began this project I not only wanted to understand these sets, I wanted to explain them to others. I believe lots of people could take an interest in these structures and the math behind them because they appeal to the artistic side of people and feed their curiosity. Mandelbrot and Julia sets provide a way for people to become involved with math. I believe my project demonstrates a way to understand Mandelbrot and Julia sets, not just as beautiful pictures, but as a fun and interesting branch of mathematics.

Summary Statement

By learning about trigonometry and complex numbers I was able to better understand and appreciate the inner workings of the fractals.

Help Received

My dad helped me write Java programs to visualize fractals. He also helped me with my poster.

Name(s)

Adrian S. Derderian

Project Number

J1502

Project Title

The Wisdom of the Crowd

Abstract

Objectives/Goals

The effects of the Wisdom of the Crowd can be measured if the sample size of the group is large enough. An experiment was conducted using various sample sizes of people who guessed the amount of jellybeans in a jar. If the people's guesses in one of the larger sample sizes were averaged, it would be likely that the average guess would come within the close vicinity of the actual amount. However, if a smaller sample size of people's guesses were averaged, the results would not come as close compared to that from the larger sample size.

Methods/Materials

Materials: Jar, Unknown amount of jellybeans, 81 participants, Ballots.

Procedures:
1. Put an unknown amount of jellybeans in a jar.
2. Create a ballot asking for guess, age, and gender.
3. Choose a sample size of people and ask them to fill out the ballots.
4. Collect the ballots and enter the data into Excel.
5. Partition the people's guesses using 5 groups with sample sizes 7, 17, 27, 37, and 47. Do this 10,000 times.
6. Plot the results and draw conclusions.

Results

A Java program was coded to randomly select guesses from an 81 sample size. It was inconvenient to repeat this experiment multiple times because another 81 people would have to become participants for each new trial. To solve this, the Java program performed random selections of people 10,000 times into the 5 groups. This is called bootstrapping. The five groups that were bootstrapped were 7, 17, 27, 37, and 47. The largest sample size, 47, only had an average error of 188 jellybeans. However, the smallest sample size, 7, had an average error of 354 jellybeans. This confirmed my hypothesis.

Conclusions/Discussion

In conclusion, this project confirmed my hypothesis that as the sample size increases, the error between the average guess and the actual amount decreases. This is because as a large amount of data flows in, the overestimates and underestimates cancel each other out producing a result that is near the actual amount. Collecting data from diverse groups of people will help cancel noisy outliers in their guesses.

Summary Statement

The accuracy of the combined independent opinions of a group increases with the size of the group.

Help Received

My Java teacher Dave Dunn taught me how to read from and write to a file.

CALIFORNIA STATE SCIENCE FAIR
2017 PROJECT SUMMARY

Name(s)

Brooke A. Dombkowski

Project Number

J1503

Project Title

Switch Craft: Is the Switch Strategy Still Optimal When the Monte Hall Game Is Expanded to More Than Three Doors?

Abstract

Objectives/Goals

It is well established that a "switch strategy" gives you a better chance of winning a prize in the traditional 3-door Monte Hall Problem. The objective of this study was to see if the "switch strategy" would still be preferred if you played the game with more than three doors. Additionally the goal was to see how the statistical chances of winning would change and could be predicted.

Methods/Materials

Cups, toy goats, toy car, computer, Microsoft Excel. Wrote a code to simulate playing the game on a computer automatically. Use this to show the results from playing the game 10 times, 100 times, 1000 times etc. up to 1,000,000 times.

Results

The "switch strategy" gave you a better chance of winning the prize, even when the game was played with more than three doors.

Conclusions/Discussion

I predicted a formula for the chances of winning if you played with "n" doors and always switched. That formula is (n-1)/n.

Summary Statement

I simulated millions of rounds of the famous Monte Hall game with various numbers of doors then proposed a formula for the chances of winning if you switch.

Help Received

Jennifer Parker, Bryn Dombkowski, Ashley Dombkowski

CALIFORNIA STATE SCIENCE FAIR
2017 PROJECT SUMMARY

Name(s)

Joshua A. Gallegos

Project Number

J1504

Project Title

Cracking the Code: Determining the Strength of Passwords

Abstract

Objectives/Goals

The purpose of my science experiment is to find out what type of password is the strongest. The reason why I am doing this project is because these days you use passwords on a daily bases; from social media to bank accounts. However, we have so many passwords we might be inclined to make them all the same or simple. We forget how sensitive this information is and the risk we take if it falls into the wrong hands.
I will use passwords with numbers, letters, and a combination of the two. To get these passwords, I will ask 30 people to make up a password, 10 people will be in each group. I will also use a password strength detector to find the strength of the passwords. I will then find the average strength of each password and compare them.
The control group will be the passwords with both letters and numbers.
After I am done with my project, I expect to find the strongest type of password to keep my accounts safe.

Methods/Materials

Die, Computer, 30 participants to come up with one password each and website http://password-checker.online-domain-tools.com/ to test each password. I did not modify the website.

Results

After completing my experiment, I concluded that Group 2 (passwords with numbers and letters) had the highest average strength of 51%, followed by Group 1 (passwords with only letters) having an average strength of 48%. Group 3 (passwords with only numbers) had the smallest average strength of 28%. These
results tell me that the strongest passwords (according to my experiment) are the passwords with numbers and letters, followed by only letters, and passwords with only numbers is the weakest.

Conclusions/Discussion

After my test I concluded that the passwords that have both letters and numbers are the strongest of the three passwords. Furthermore the passwords with only letters are stronger than the passwords with only numbers.

Summary Statement

By testing the strength of various passwords using an online source, i was able to determine the strongest type of passwords.

Help Received

None, the only outside help I used the the online website password checker to determined the password strength.

Name(s)

Andrei Mandelshtam

Project Number

J1505

Project Title

Limiting Behavior of the Iterations of Tangent

Abstract

Objectives/Goals

The objective of this project was to study the statistical properties of the iterations of tan(x).

Methods/Materials

Laptop computer with Fortran compiler and Grace. Ran program on 100 initial values of x, iterating tangent on each one 10 million times, and distributed the iterations in a histogram.

Results

Two new theorems were proven. The first theorem showed that the iterations of tan x have no limit unless some iteration falls directly into a solution to tan(x)=x, this being only a countable number of points. My most interesting theorem is that periodic points are dense on the real number line. Additionally, the inverse square law was discovered numerically by studying the distribution on a histogram of the first 10 million iterations of tan(x) for 100 different starting points y. It was found that each of these histograms closely resembled the graph of C/x^2 for a varying constant C. The values of C computed by the method of least squares showed random-like behavior. Then each histogram was scaled by its value at a fixed point a. It was found that each one resembled the same graph, a^2/x^2, no matter what starting point y.

Conclusions/Discussion

Through rigorous and numerical results, it was shown that the iterations of tangent have a very interesting structure with many unusual patterns and properties. Two theorems describing the iterations of tan(x) were proven rigorously using the methods of calculus, trigonometry, and mathematical induction. Also, it was shown numerically that the iterations have a universal, non-random distribution.

Summary Statement

I studied the iterations of the tangent function, proving two theorems and discovering a universal law their distribution satisfied.

Help Received

My father taught me programming in Fortran. My mother gave me ideas on what to investigate after I proved my first theorem. My science teacher, Mrs. Martin, gave me general Science Fair advice.

CALIFORNIA STATE SCIENCE FAIR
2017 PROJECT SUMMARY

Name(s)

Anna V. Orgel

Project Number

J1506

Project Title

The Relationship between Password Characters and Guesses Required Using a Self-Developed Brute Force Hacking Program

Abstract

Objectives/Goals

This study measured the variation in number of guesses required for different types of passwords using a brute-force program I created and modified on scratch, a computer coding website.

Methods/Materials

Materials: Laptop computer with self-developed Scratch program, online character randomizer (Random Letter Sequence Generator: www.dave-reed.com)
Used a website to generate random passwords of a given length and create lists of 25 passwords for each category containing various numbers and types of characters. Compared average number of guesses required for each type of password using self-developed Scratch program. This program was created by modifying an existing program on Scratch that determined password strength. The modifications I made were:
1) Expanded the original program into seven categories for each type of password;
2) Changed the variables and lists to match those within each category;
3) Added a counter to display total number of guesses required for each password;
4) Added function to allow user to choose designated password character category before entering password.
Number of guesses for each password were recorded and averages taken for each category. Averages were then compared to a previously predicted number based on the formula: number of possible characters^digits.

Results

The results of the study showed a positive, exponential relationship between the number of possible characters and the average number of guesses required. The passwords with more possible characters were most secure, and the passwords with fewer possible characters were less secure. The results closely matched the predicted values using the formula: number of possible characters^digits. My hypothesis was supported.

Conclusions/Discussion

Every decade we become more reliant on technology, but as we become more adept at using it, hackers threaten to compromise our security. My study and the ease with which my program guessed passwords confirms the necessity of higher level security precautions for computers other than passwords. Furthermore, it demonstrates the overall fallibility of passwords, especially given our human tendency to select those that are most easy to remember, and therefore most easy to hack.

Summary Statement

My study examined the relationship between the number and type of characters in a password and how easily they were guessed by a program that I developed using Scratch, a programming website.

Help Received

The program was developed completely independently, excluding minor bug checking help from my math tutor. The experiment was performed with no help, and the write up was only briefly edited by a parent.

CALIFORNIA STATE SCIENCE FAIR
2017 PROJECT SUMMARY

Name(s) Hunter D. Ruebsamen	**Project Number** J1507

Project Title

Evolving Near Optimal Solutions to Computationally Hard Problems Using Natural Selection

Abstract

Objectives/Goals
Determine if processes of natural selection can be applied to non-biological or simulated entities to produce nearly optimal solutions to a variety of computationally hard problems in a reasonable amount of time.

Methods/Materials
Desktop computer with Processing Compiler/IDE, Processing.js extension and Chrome browser. I developed my own algorithms using research on the internet, for my 3rd problem of evolving locomotion in virtual agents, I borrowed from some open-source code published by Cary Huang and extensively modified it.

Results
Three different problems were easily solved using the algorithm I devised. Results show nearly optimal solution to Linear Regression Problem in as little as 5 generations and Traveling Salesman Problem of 350 cities produced near optimal results after 200 generations and evolving locomotion in virtual creatures produced good results within 300 generations of evolution.

Conclusions/Discussion
I developed an algorithm based on natural selection that can be used to find near optimal solutions to a variety of complex problems in a reasonable amount of time. I show that it is not only possible for simulated evolution to work with non-biological entities, but also demonstrate that evolution can produce solutions to computationally complex problems that are difficult or impossible to solve using traditional methods or exhaustive search.

Summary Statement
Using simulated evolution to generate near optimal solutions for computationally complex problems

Help Received
I learned about Linear Regression in my Algebra class. I researched the Traveling Salesman Problem online and the code for my locomotion of virtual creatures was based on open source code by Cary Huang and modified by myself. The simulated evolution algorithm was programmed by myself after research on

CALIFORNIA STATE SCIENCE FAIR
2017 PROJECT SUMMARY

Name(s)

Jonathan Z. Xu

Project Number

J1508

Project Title

Analysis of Maze Solving Algorithms

Abstract

Objectives/Goals

Find the most efficient maze-solving algorithm among three different algorithms: the #right-hand rule#, #dead reckoning#, and #dead end remembering#.

Methods/Materials

The programming language I used is called Java. I developed all the algorithms in an Integrated Development Environment (IDE) called Processing. I downloaded open-source code from GitHub for random maze generation. I programmed different maze-solving algorithms. Then I ran 8 tests, each measuring how many moves taken by each algorithm on the same maze. The maximum number of moves, minimum number of moves, and average number of moves by each algorithm among all 8 tests are recorded. The code generates a random maze each test so that a variety of mazes with different possibilities are covered.

Results

Right-hand rule kept a consistent range (178~397), as well as having the least average (293.87). Dead end remembering was better than dead reckoning, with less major outliers.In test #2, the maze generated had many forks, which resulted in the two random algorithms (#dead reckoning# and #dead end remembering#) having many more moves. However, in test #8, dead end remembering takes only 43 steps to reach the center comparing to more than 200 steps by the other algorithms.

Conclusions/Discussion

I believe dead end remembering was still the most efficient out of the three. While the maze generation code was programmed not to generate loops, the right-hand rule would be stuck in an infinite loop, but eventually the random algorithms would be able to solve the maze. While dead reckoning is too pointless, getting stuck in long loops over and over, dead end remembering was made to improve on this. It will not be stuck in a loop for too many times due to its mental wall creation technique, and will efficiently solve any maze.

Summary Statement

I programmed 3 different algorithms in Java and I tested them on 8 randomly generated mazes to determine the most efficient algorithm.

Help Received

I downloaded open-source code from GitHub for random maze generation.

CALIFORNIA STATE SCIENCE FAIR
2017 PROJECT SUMMARY

Name(s)

Richard T. Zhu

Project Number

J1509

Project Title

Open-2-Win: Are Popular Chess Openings Really Better?

Abstract

Objectives/Goals

The objective of this experiment was to test what effect popularity would have on the results of real-world games, as popularity is an indirect factor in an opening's success.

Methods/Materials

I used a device with access to the Internet such as an I-Pad or a laptop to go to the chess.com database. Then, using database information collected by the database over thousands of master games, I wrote down the statistics and compared them with the popularity of the opening and types of openings.

Results

I found that usually the popular openings were usually better for the side initiating them though usually white dominated due to him having the first move. This agreed with my hypothesis overall predicting that popular openings for a side would have better results for it-why would it be played?

Conclusions/Discussion

This project shows us how usually if an opening is studied, there will be new lines opened up, etc., and how people play openings for hopeful future profit from them. I concluded that popularity indirectly influences openings to be better for the initiating side. This project can help people clarify which openings they would probably want to pick, providing an overall view of the several most played openings on the spectrum. More studies can be done to determine relations between other factors and success rates, such as king safety, etc.

Summary Statement

As popularity is an indirect factor in an opening's success in real games, I tested what effect it would have on the results of real-world games.

Help Received