VISUALIZATION OF ONLINE ENCYCLOPEDIA OF INTEGER SEQUENCES

HTTPS://GITHUB.COM/DAVIDJMILLER/VOEIS

Qianlang Chen (u1172983)
u1172983@utah.edu

David Miller (u1312141)
u1312141@utah.edu

Jiawen Song (u1211977)
kevin.song@utah.edu

1 INTRODUCTION

We propose to visualize the integer sequences found at the Online Encyclopedia of Integer Sequences (OEIS). The OEIS is the largest database of integer sequences, ranging from mathematically important sequences, such as the Fibonacci sequence and Collatz sequence, to fun sequences, such as the topologically-distinct pizza slices sequence. Within this collection of integers, mathematicians believe there to be beautiful patterns hidden. These patterns often explain things in nature, such as the ratio of adjacent Fibonacci terms converging to a constant found in ocean waves and flower petals, or interesting and mesmerizing patterns, such as the Ulam spiral or Thue-Morse turtle paths. It is clear that visualization is a useful tool to probe into the world of numbers to undercover these patterns, and with such a large dataset that is constantly growing, it lends itself well to being put within a computational framework that can handle this.

2 BACKGROUND AND MOTIVATION

We choose this project because we are naturally curious about numbers. Naturally one would think some numbers have more importance than others, such as primes or perfect squares having high significance. In fact, this is what N.J. Sloane found when he plotted an integer n versus its number of occurrences in all sequences $N(n)$. This led to what is knows as Sloane's Gap, a noticeable gap between two bands in the plot where "significant" numbers tend to appear on the upper band as can be seen in figure 1. It is this moment of "Wow!" that really drives this project. Finding importance in numbers is the basis of Number Theory and this project.

3 PROJECT OBJECTIVES

We hope through visualization that we can uncover why certain numbers are important. In fact we hope tho answer a myriad of questions other than this one:

- Do certain numbers appear in certain positions of sequences?
- Given two sequences, can we relate them somehow?
- What does the way a sequence progresses say about it?
- Which numbers are important to their sequence (local importance) while which numbers are important as a whole (global importance)?

We are amateur, enthusiastic number theorists that just want to probe and investigate how numbers manifest themselves through sequences. We are walking along the landscape of integers with the hopes of finding patterns using visualization as our compass.

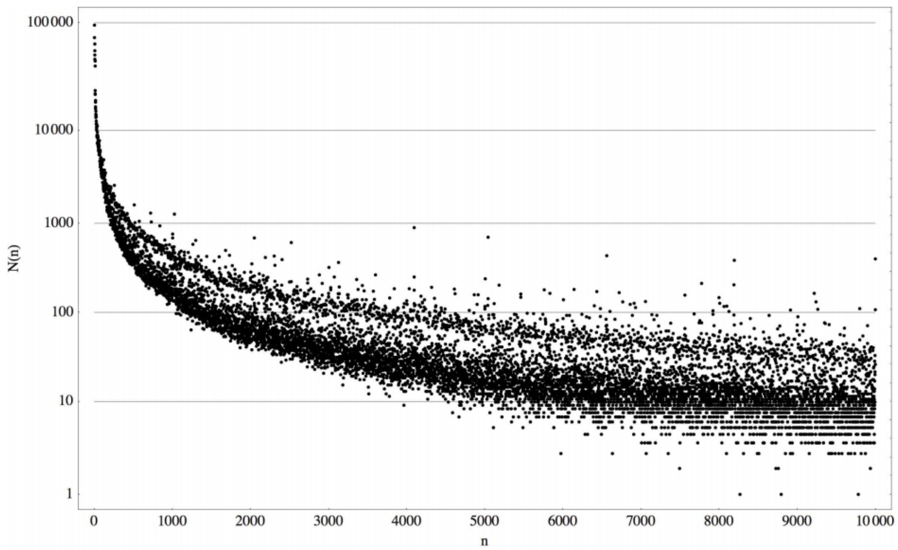


Figure 1: Sloane’s Gap, depicting two distinct bands (Gauvrit et al. (2011)).

4 Data

There are three forms in which data is available to us: names, stripped, and JSON. names and stripped are .txt files that are provided by the OEIS Foundation in compressed files found at `https://oeis.org/wiki/Welcome#Compressed_Versions`. The JSON data of each sequence can be easily accessed by curling `https://oeis.org/search?q=id:AXXXXXX&fmt=json`, where we would replace `XXXXXX` with the appropriate A number.

1. **names** is of the regex form: ”(A[0-9]ˆ6)\ (.+)”, where the first group is the A number of a sequence, or its ID in the database, and the second group is the sequence’s name. The first few lines in this file are comments and start with ”#”. Here is an example of a couple of lines from names.txt:

   ```
   A000059 Numbers n such that (2n)^4 + 1 is prime.
   A000060 Number of signed trees with n nodes.
   ```

2. **stripped** is of the regex form: ”(A[0-9]ˆ6)\ ,((-?[0-9]+,)+)”, where the first group is the A number of a sequence, and the second outer group is the actual integer sequence. The first few lines in this file are comments and start with ”#”. Here is an example of a couple of lines from names.txt:

   ```
   A000059 ,1,2,3,8,10,12,14,17,23,24,27,28,37,...
   A000060 ,1,2,3,10,27,98,350,1402,5743,24742,...
   ```

3. **JSON** data has many fields, including: references, formulas, code to generate the sequence, and time of creation. There are many more fields within the JSON of each sequence, but the ones listed are the ones that are of use to us.

5 DATA PROCESSING

All the data required for our project is already given to us in a simple, clean format explained in the previous section and therefore do not anticipate any data cleanup. However, given the objectives described above, we will need efficient data structures to service client requests. Some data structures we are considering are

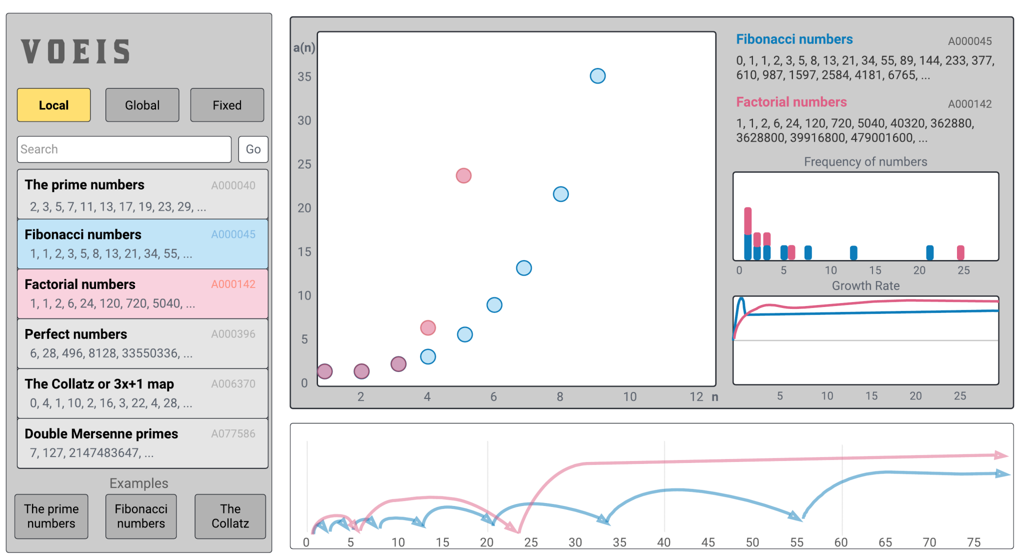
- **Sequence name lookup**: A data structure mapping A number to the sequence name and actual sequence.
- **Integer lookup**: A data structure mapping integers to a dictionary that contains tuples of the A number of the sequence they appear in and the positions within that sequence.

The current idea for the data structures is just dictionaries stored in RAM using Python. However, if we find this to be too inefficient, we will transition to SQLite as it has nice integration with Flask. We are also considering the following metrics for our data:

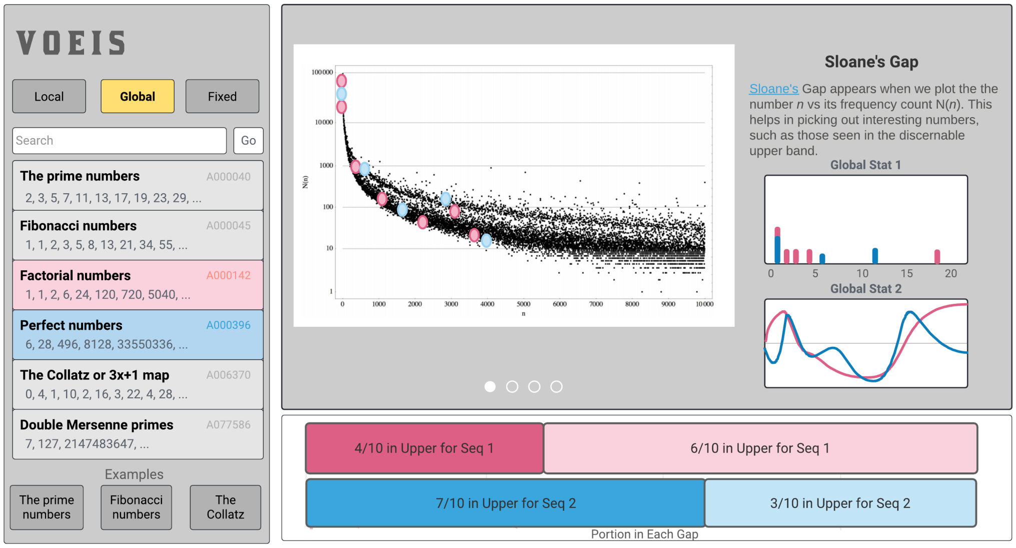
- **Integer distance(**$input\_seq$**)** so that we can return the top n results if a user inputs an arbitrary sequence. That is, we will match a user's query to the closest integer sequence available within the database. We will either try and use the rankings provided by OEIS or compute them our self based on how OEIS ranks their queries (Sloane & Cox (2015)).
- **String distance(**$input\_str$**)** so that we can return the top n results for sequences that have relevance with $input\_str$.

6 VISUALIZATION DESIGN

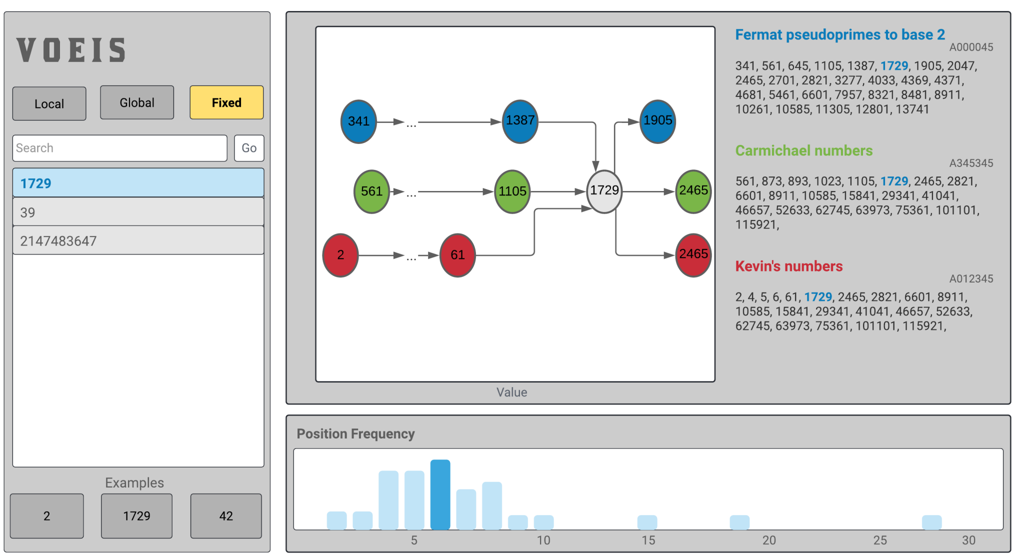
6.1 MAIN VIEWS



The Local View shows information about the selected sequences themselves. The main scatter-plot in the center plots the selected sequences directly. The arrow chart on the bottom shows animation about how the sequences grow. The plots to the right of the main plot show additional statistics about the sequences. All charts are encoded with a different color for each selected sequence. Moreover, the user can hover over any chart to change the mode of visualization (changing from a bar chart to a line chart, for example), or to show a different stat.



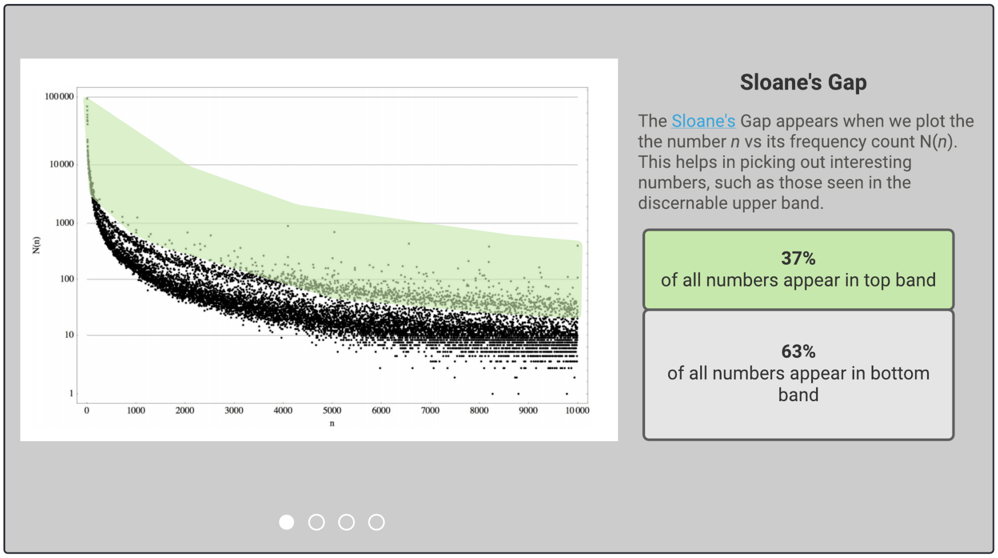
The Global View shows information of the selected sequences when compared to all sequences in the database. The main chart in the center highlights the elements in the selected sequences from all numbers. The plots on the right show more useful global stats about those sequences. Since all elements from all sequences are separated by the famous Sloane's Gap, it would be interesting to know if a sequence is made up from mostly "popular" numbers. The bottom chart helps with this by showing the proportion of "popular" elements in each selected sequence.

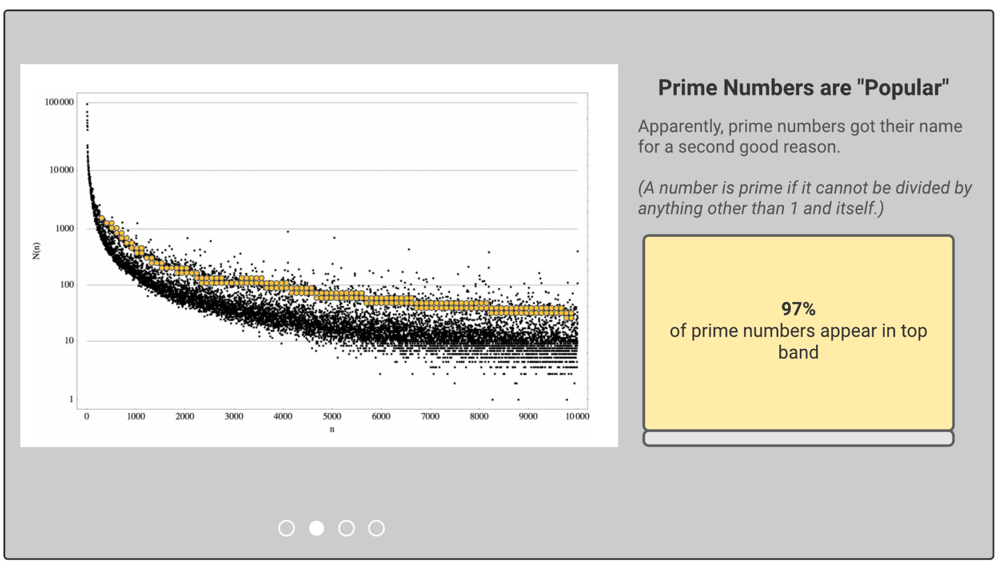


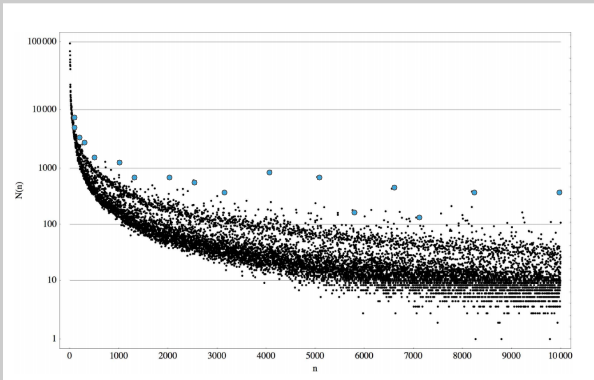
The fixed views visualizes the role of a specific integer with the context of sequences. The large graph will have a visual representation of how an integer is represented within sequences. This means we should be able to investigate its neighbors within sequences, where it usually falls within a sequence (position frequency graph), and its multiplicity. This view will allow us to determine if a number is important or not.

6.2 STORYTELLING

We have four views for storytelling (Figure 5 - 8). Specifically, we insert these four views under the Global View to show the overall characteristics of the Sloane's Gap, the prime numbers, the high composite numbers, and small numbers to pitch audience interest into the number sequences. In the global view, if you scroll down, you will get access to these four views.





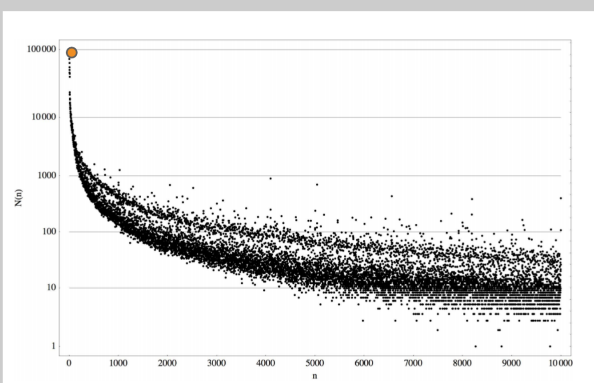


Highly Composite Numbers are "Highly Popular"

These anti-prime numbers are rare and are also real gems for mathematicians.

(A number is highly composite if it has more divisors than any number smaller.)

91%
of highly composite numbers appear in top band



Small Numbers are "more interesting"

Our "everyday-numbers" are indeed "every-sequence-numbers."

79% of all sequences cointain at least a number less than 10.

The number 2 alone appears in **33%** of all sequences and is the "most popular."

7 Features

7.1 Must Have Features

For this project, there are several must have features:

1. **An efficient search bar** in which the user can query a sequence and we return the sequence they search for, or the closest match if the input is not exactly a recorded sequence, and the sequence number. With this, we would also want a history of sequences looked up so that the user can compare sequences.
2. **A local view** that visualizes an integer sequence by itself or against a select few the user selects from history. This is important to investigate a sequence in detail.
3. **A global view** that visualizes an integer sequence against all other sequences and aggregate data. This is important to investigate a sequence and how it relates to all other sequences.
4. **A fixed view** that visualizes an integer and how it is represented within the OEIS. This is important to investigate the significance of an integer within the context of sequences.
5. **Click and hover events** where we highlight the clicked/hovered sequences within the graph.

The last three are context in which we find it to be pivotal to investigate the data. The actual visualizations are an open-ended question.

7.2 Optional Features

For this project, there are several optional features:

1. **Storytelling of Sloane's gap** where we have multiple views of important numbers in Sloane's gap. This will be a nice feature to have since it gives users who are not familiar with integer sequences a gentle introduction as to why we care about integer sequences.
2. **Synchronous transitions** where each graph within the same page transitions the same. That is, when the i-th integer in sequence is plotted in one graph, so it is in the others.
3. **Click and hover events** where we add more substantial triggered events, such as looping how a sequence progresses within the local view.
4. **Works with all zoom levels** because zooming messes with the views.
5. **Brush event** to focus on a restricted view so the user can investigate areas of interest.
6. **String lookup** where the user inputs a string describing a sequence and we return the top n results.

8 Project Schedule

Our project can be broken up into two main programming challenges: the server backend and the client fronted. The former includes using Flask and data structures to serve client requests and efficiently index the data to retrieve whatever the request is asking for. The latter programming challenge involves much of the work we have done throughout homeworks. In fact, this is more of a creativity challenge than a programming challenge as how we will visually display integer sequences is a problem that has many aspects to it. Therefore we have two main challenges: programming a server-client model and creating a creative visualization for our data. With this in mind, we have the following tentative schedule:

Dates	Goals
Oct 30 - Nov 6	· Setup client-server model to handle both POST and GET requests · Setup data structures mentioned earlier · Have the homepage with the grid layout
Nov 7 - Nov 13	· Must have features with at least one visualization per graph
Nov 14 - Nov 20	· Start on optional features · Implement more visualizations for each graph · Nov 15 is milestone meeting, must have functional prototype
Nov 21 - Nov 27	· Work on any feedback from milestone meeting · Final testing, implement more features
Nov 28 - Dec 2	· Clean up code and document

We understand that the goals setup within the first week above are not within the scope of this course, and that is why if we can not achieve a working beta of a server-client by this point, we will concentrate on just a client side data visualization.

REFERENCES

Nicolas Gauvrit, Jean-Paul Delahaye, and Hector Zenil. *Sloane's Gap: Do Mathematical and Social Factors Explain the Distribution of Numbers in the OEIS?* 2011. doi: https://arxiv.org/pdf/1101.4470.pdf.

N.J. Sloane, 2009. URL `https://oeis.org/`.

N.J. Sloane and R. Cox, 2015. URL `http://oeis.org/wiki/How_Replies_Are_Ranked`.