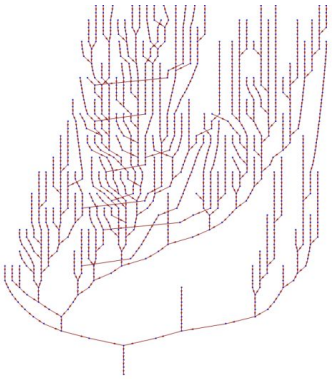
VOEIS Process Book

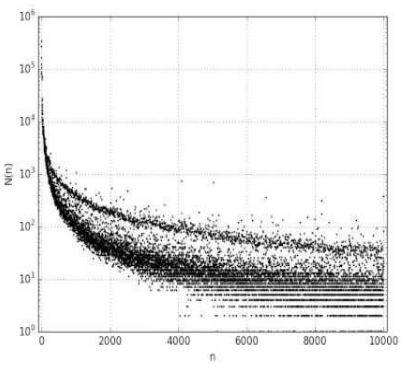
https://github.com/DavidJMiller/VOEIS

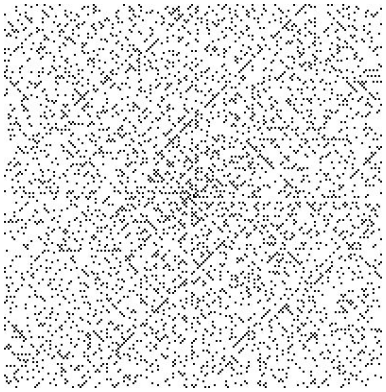
Qianlang Chen	David Miller	Jiawen Song
u1172983@utah.edu	u1312141@utah.edu	kevin.song.utah.edu
u1172983	u1312141	u1211977

Overview and Motivation

The importance of numbers is a core principle in Number Theory. Whether studied in an isolated manner or within the context of sequences, numbers contain patterns and significance. While Number Theory aims at uncovering these patterns and significance with mathematical tools, we propose to use visualization as our probing tool. It is no doubt that visualization can lead to interesting images that seem to show these hidden patterns. The Ulam Spiral, Sloane's Gap, and the Collatz Tree are a few that have been popularized through *Numberphile*, a YouTube channel that explains interesting patterns in math and a source of inspiration for us.



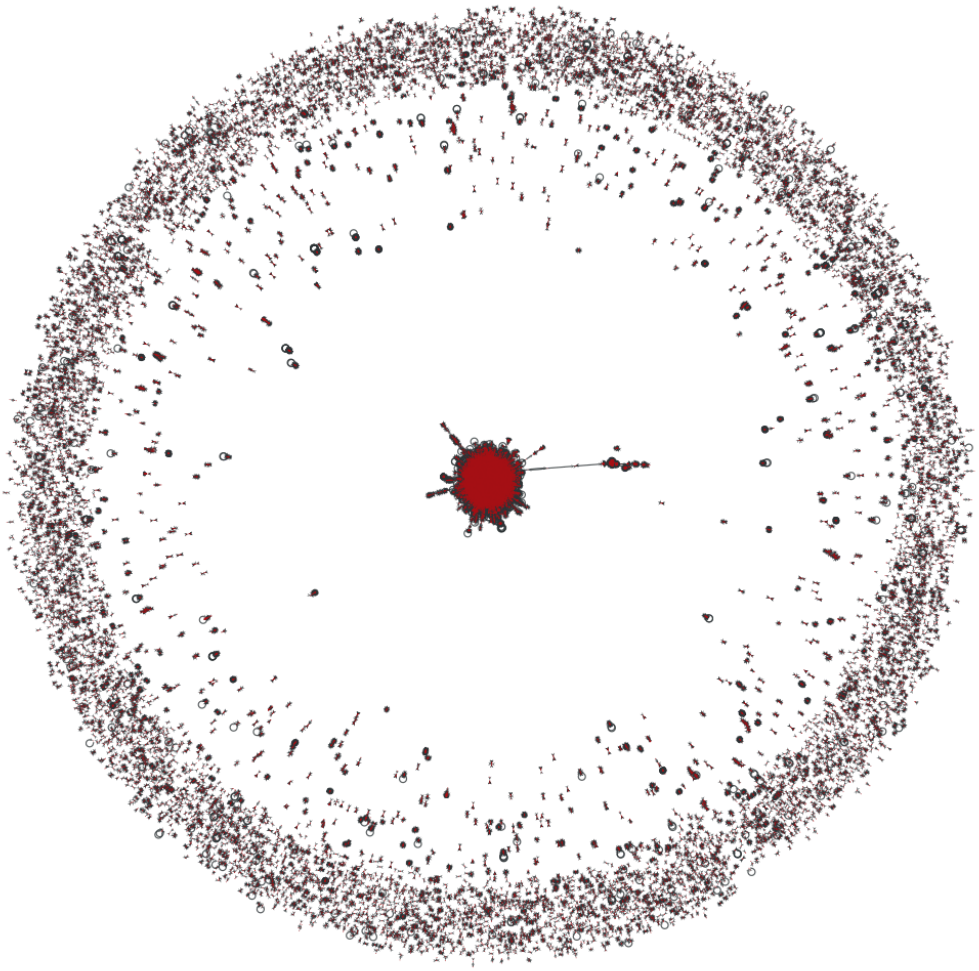




Since we are very interested in these patterns, naturally we asked ourselves what is the relationship among all patterns. We really want to determine why certain sequences spawn interesting visualizations while others do not or why a certain number contributes to many sequences. The goal of this project is to explore sequences and numbers within the context of sequences through visualization. Specifically, we are interested in visualizations that can convey meaning and are pleasing. Standards graphs and charts conveys the first part while the above visualizations conveys the second. We aim to create visualizations that convey both parts.

Related Work

There are many sequence visualizations that exist, as seen in the previous section. It is these interesting standalone visualizations that first sparked our interest. These first inspirations were encountered through *Numberphile* videos. Other inspirations are Dr. Katherine Stange's *Experimental Mathematics Lab*, in particular their project *Numberscope* (https://math.katestange.net/numberscope/), the discovery of new sequences through graph visualization in the paper *Integer Sequence Discovery from Small Graphs* (https://arxiv.org/pdf/1408.3644.pdf), and independent projects such as the one found at https://github.com/internaut/d3-int-sequences-visualizer. It is these visualizations which we drew inspiration from to create visualizations that are not only pleasing, but also informative. One thing we notice though is that most of the visualizations are a local viewpoint, that is the visualize the sequence within sequence within their own context. However, we are also interested in comparing sequences to each other as well as investigating the role of specific numbers within the context of sequences. One visualization, which can be found at https://sam.zhang.fyi/2018/12/01/oeis/, shows that there might be an inherent structure to how these sequences interact with each other. Sequences on OEIS have cross references to other sequences and the above graph shows sequences as nodes in which an edge connects two nodes if there is a cross reference between them. The graph below shows there is some inherent structure between sequences.



With the core idea of being able to visually investigate not only sequences, but multiple sequences and numbers, we plan to make visualizations from the mentioned inspirations.

Questions

What questions are you trying to answer?

- The questions we are curious to get answer from making this project is to explore the number sequence in the visualized way to find the patterns of each specific sequence and its relations to other sequences.

How did these questions evolve over the course of the project?

- Over the course of this project, we can see that there are a number of ways to visualize sequences that can be very intuitive for the people who are not exposed to number theory or math in general who can use this project to see the patterns.

What new questions did you consider in the course of your analysis?

- So the new questions we are considering now is how do we make users who are not just mathematicians and number theorists explore the number sequences with our VOEIS project with the help of interesting, interactive visualizations. So if the user can easily navigate to different number sequences and explore the differences and similarities among them.

Data

The Database

We've collected our data straight from the OEIS (https://oeis.org/wiki/Welcome). The raw data included two text files: one maps each sequence's unique ID to the sequence's name, and the other maps the IDs to the sequences' terms. We decided that we'll keep all relevant data in the Python server's local memory as dictionaries because that allows for faster lookup times. To make the process of building those dictionaries easier, we reconstructed the raw data from OEIS and called them our "database." Since our website has functions of showing information about sequences and numbers, we thought it'd be nice and friendly to our server if made the following databases:

- A **sequence-database** that maps a sequence's unique ID (officially called the "A-number" of a sequence) to its name and terms:

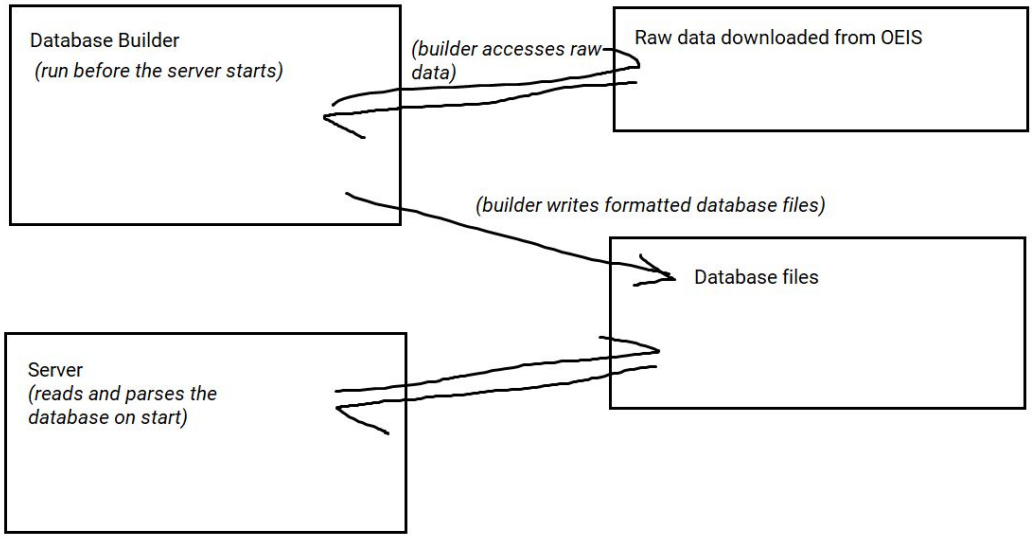
a_num	**name**	**terms**
"A000040"	"The prime numbers"	2, 3, 5, 7, 11, 13, 17, 19, 23, 29, ...
...		

- A **number-database** that records useful information about a number that has ever appeared in an OEIS sequence. A number-data item contains the following fields:
    - The total number of appearance of this number;
    - The total number of OEIS sequences this number has appeared in;
    - The total number of appearance of this number at a particular index of any sequence;

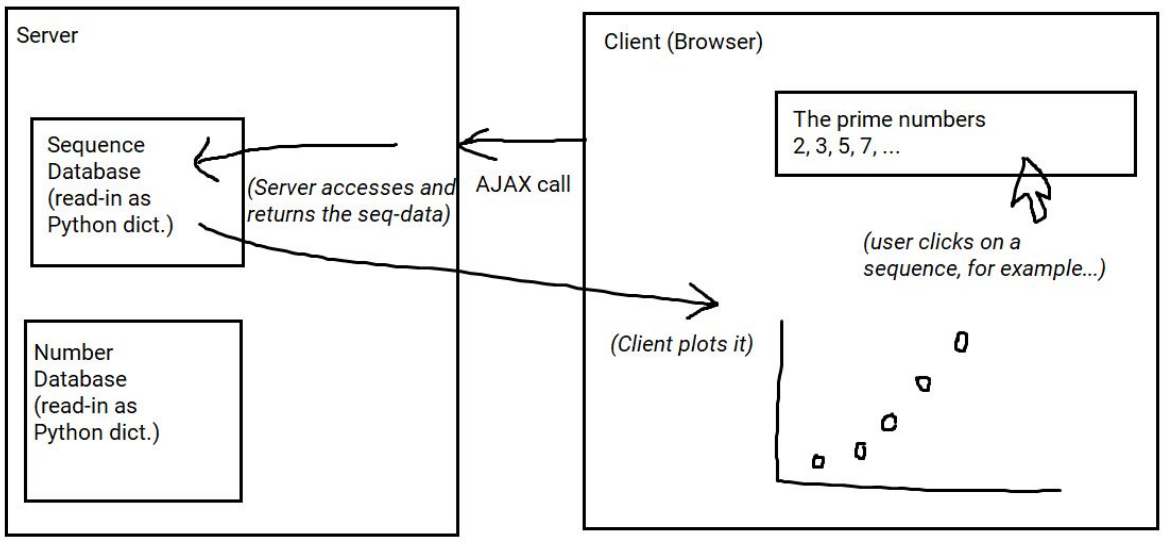
- The “popular neighbors” of this number. A neighbor is another number that likes to appear near this number in any sequence. Our database will include the neighbors of each number within six indices apart.

num	**total_count**	**total_seqs**	**index_counts**	**neighbors**	
				offset	**neighbor_counts**
2	312953	139205	{1: 4245, ...}	-6	{1: 3434, 3: 3565, ...}
				-5	{1: 3569, 2: 520, ...}
				...	
...					

These two database files are well-formatted text files that allows our server to quickly read and parse them as soon as it starts. This way, our server doesn’t need to read the files when the client’s already asking data, nor does it need to perform complex computations when certain data is asked as when the server only has the raw data at hand. We’ve included the Python program that does the database building process in our project, so we can download updated sequence-data from OEIS and rebuild our database at any time.

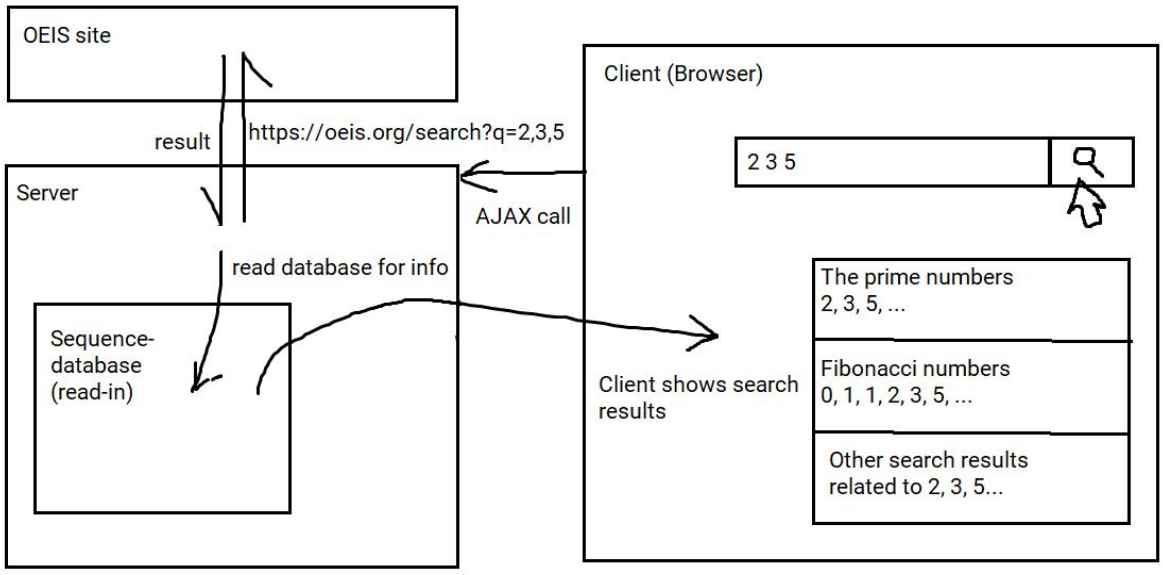


We use AJAX (Asynchronous JavaScript and XML) to communicate between the client (browser, JavaScript) and the server (Python). Every time the user chooses to lookup some sequence or number, the pertinent sequence's A-number or the number are sent as the input of the AJAX call. Our server will then lookup our read-in database, which are stored as Python dictionaries in the server's RAM and have high-performance lookup, and return the requested data.



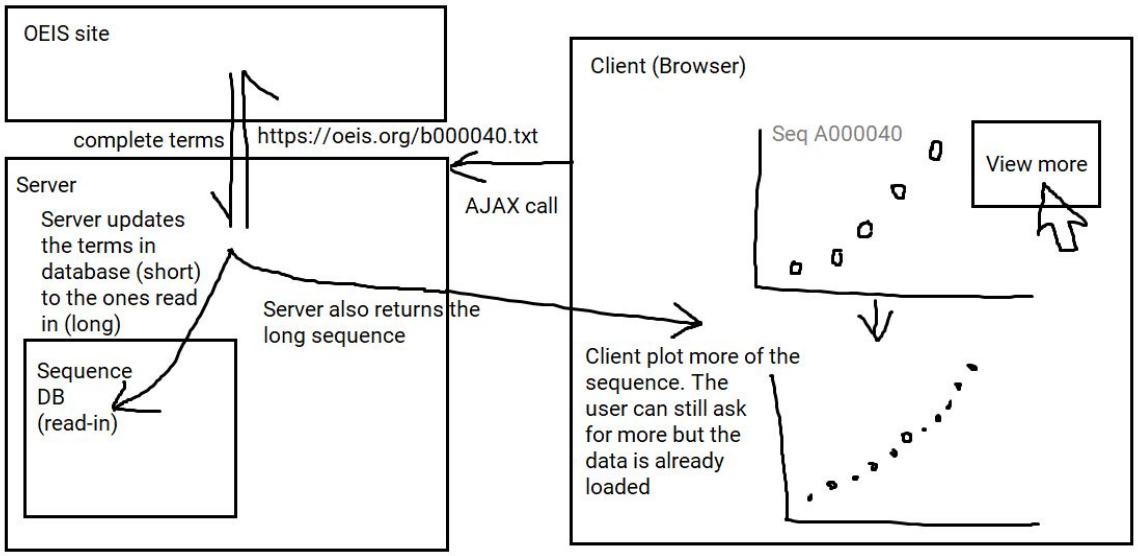
The Search Function

We've made use of the search function from the official OEIS site (https://oeis.org). The official OEIS' search function allows queries of sequences by both the names and the terms, which is very useful. Since sending HTTP requests to the OEIS doesn't take long, our server upon receiving a search query and internally accesses the OEIS for the search result data since OEIS has such an amazing search/ranking algorithm:

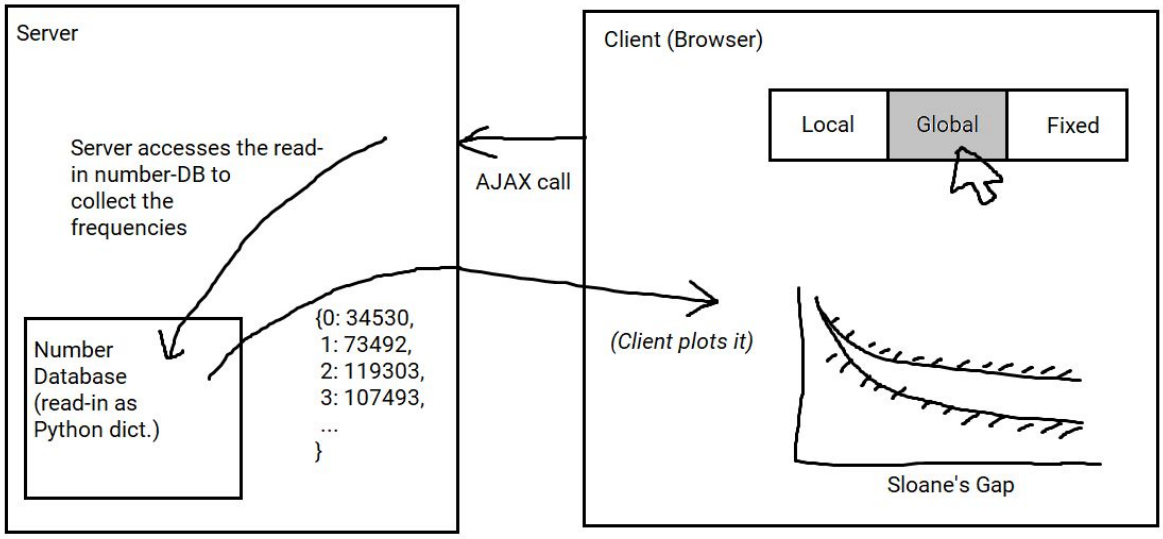


Other Related Data

The OEIS, by default, has only included the first few dozen terms (maximum) for each sequence. This is a nightmare for “prime-number enthusiasts” like us if we can only see up to the first sixty prime numbers. Actually, OEIS does have more terms recorded for each sequence (up to 10,000 terms) but they just stored them in separate text files. However, this allows the users to view more of a sequence if they want to: the client makes an AJAX call when the user’s requesting more terms of a sequence, and the server downloads the data from OEIS and updates the local database:



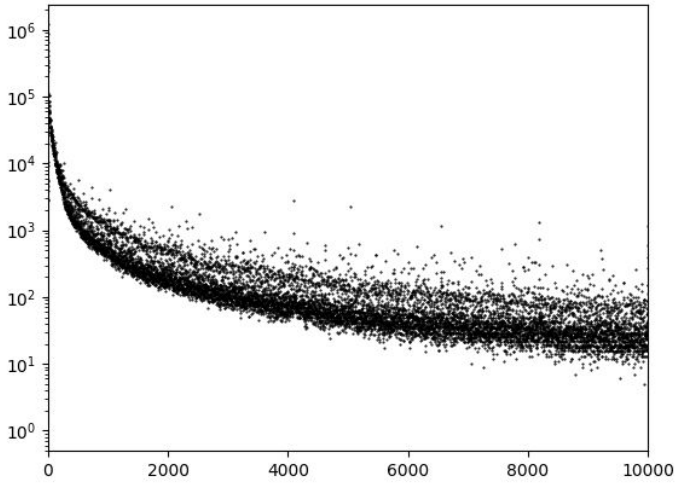
In our global-view, we have our main plot showing the Sloane's Gap, where every data point represents the number of sequences a number has appeared in; for example, the number 2 has appeared in 139205 sequences, and that'd correspond to the a dot located at (2, 139205). When the user switches to global-view, the client will just send an AJAX to request the frequency of each number, which the server will then return:



Exploratory Data Analysis

What visualizations did you use to initially look at your data?

- The Sloane's Gap is our initial way of looking at the data, which is also one of the inspirations for us into this project. It's a simple scatter plot that plots each number to the number of OEIS sequences that number has appeared in:



What insights did you gain?

- Clearly, this indicates that not all numbers are "born equally," or at least, they all get different amounts of attention from the mathematicians. In this sense, some numbers are more interesting than others.

How did these insights inform your design?

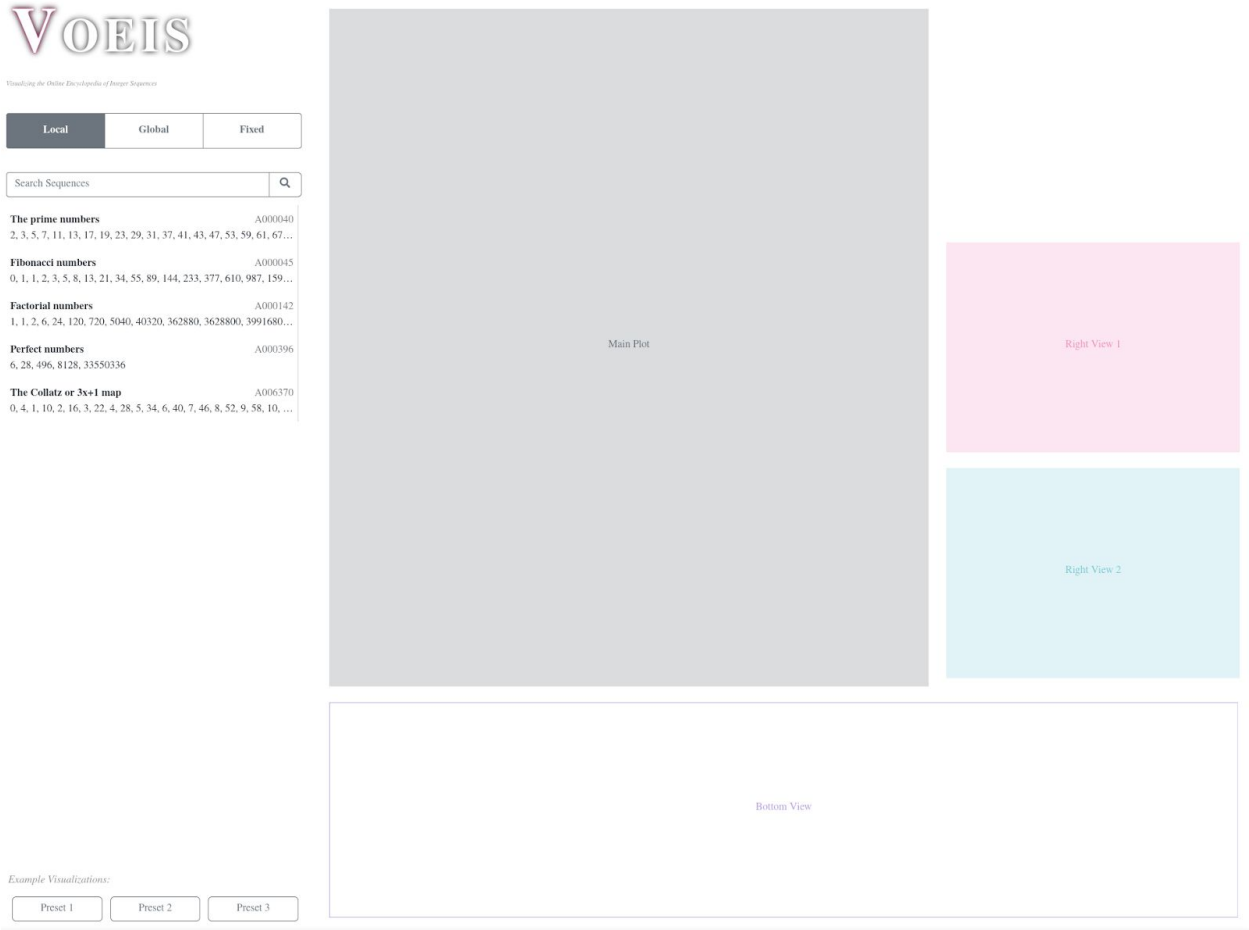
- This inspired us into visualizing the sequences in much more different ways because we, or anyone upon seeing the Sloane's Gap, would like to know what features numbers have to make them interesting. This also reminded us to come up with more statistics about the numbers to visualize to have a chance of understanding this pattern.

Design Evolution

Homepage Evolution

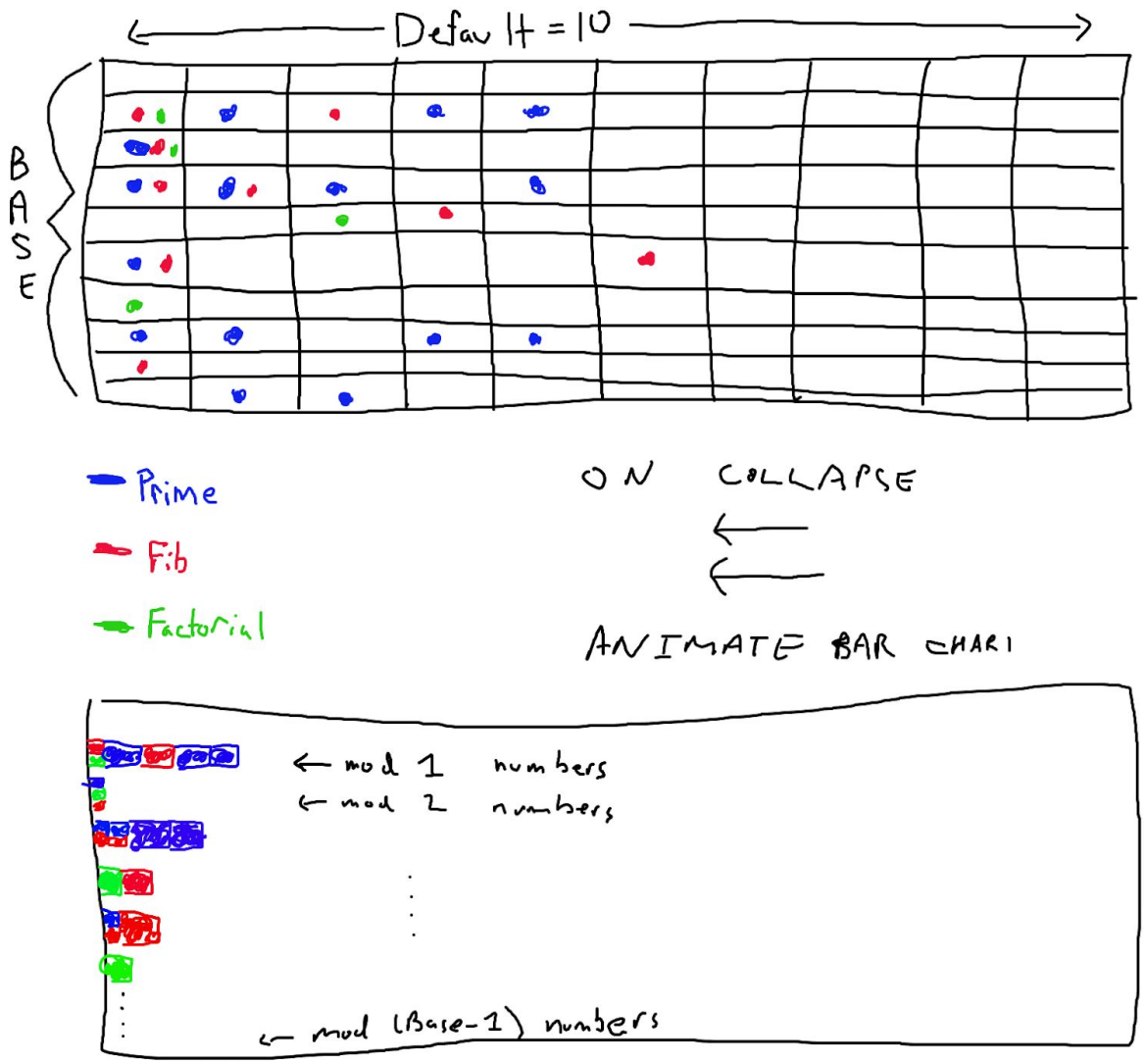
(view Iteration 1)

At first, we draw the basic layout of the three components of the VOEIS homepage, which consists of a search bar, main view, and secondary view. And then, we populate each component with a basic framework that requires no data such as the bottom to switch among local, global and fixed.



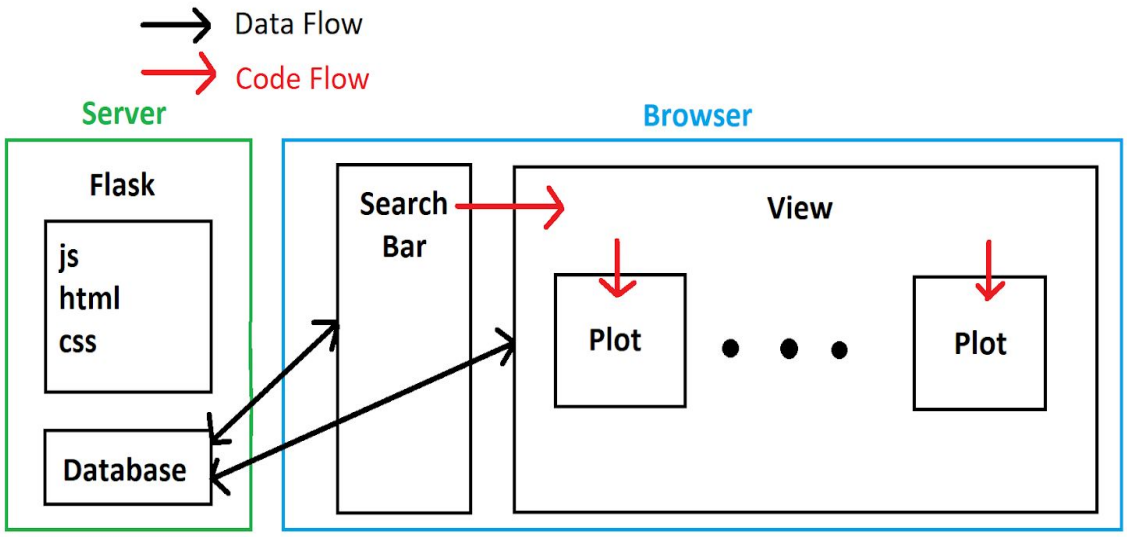
(view Iteration 2)

In the iteration 2, it has the search bar view being implemented with title, and the buttons for local, global and fixed. And it is ready to implement the various different visualizations in the main view and the secondary view.



One of the ideas for the visualization of the global view is to have a matrix table, and the number of cells is the base. For instance, if we have a base-10 sequence, we would construct a matrix table with 10 rows. And each cell will be populated with different number sequences in that value differentiated by the different colors. And to make the view more interactive, we also offer the option to collapse the matrix table by turning the matrix into a horizontal bar chart so we can observe the pattern of the different number sequence with the help of colors.

Implementation



For the browser part, our project is divided into search bar, view, and plot demonstrated above. So far, we've implemented a prototype with just scatter-plots. But we have made communication protocols for the server and the browser fully functional. The search bar is fully functional. After the milestone, we will be working on sending the data dynamically from the search bar to the view, and from the view to the plot. And we will be implementing the various kinds of visualizations that have been discussed so far.

Evaluation

From our visualization, the user can find patterns within certain sequences by using our local-view, or see the relationship between the sequence and all sequences in OEIS by using our global-view. We think our visualizations are reasonably organized and informative so that anyone could take a thing or two away from it.