# Hashing Analysis

## Discussion of Plots

After getting the results object, I organized the data into respective x, y coordinate pairs and utilized the JavaScript library, Plotly, to generate interactive graphs. This way I could generate graphs more efficiently for different experiments by avoiding exporting data to .csv and fitting them to graphs in excel.

There are four plots, one for each experiment. Each plot has three different lines that correspond to a trial based on table size. Their values and color can be found in the legend at the top right of each plot.

The x-axis represents load factor and ranges from 0 to 1, following increments of 0.2. The y-axis represents numbers of collisions at any load factor. The range of y-values changes in some of the experiments, but it is generally around 0-50 for table sizes of 10, 50, and 100.

Every trace has data points incremented by , therefore the bigger the table size, the more results you there are to work with. Each point is connected by a line of the same color.

In trials with smaller sized tables the lines in the graph seem to be increasing linearly. However, when table sizes of 10000, 25000, and 50000 are used in the experiments, every trace clearly displays positive exponential growth with collisions reaching up to 30000.

## Effects of collision resolution scheme

The effects of collision resolution screen are clearly displayed in the experiment’s results. In both pairs of experiments, the trials that utilized the separate chaining scheme finished with less collisions than the open Addressing scheme. In the experiment with the largest hash table size (50000), separate chaining reached a load factor of 1 with around 15000 collisions while the trial with open addressing had reached up to 25k collisions

This makes sense and displays what I believe is the main advantage of using separate chaining over open addressing. Separate chaining, at any point during the experiment left more spots in the array open, as collisions wouldn’t fill up spaces that didn’t *technically* belong to the key value in its current iteration. Because of this, there would be the same number of open indices in the hash table after a collision handled by separate changing. In the open addressing trials, open indices are filling up no matter what, therefore after every single collision, the chance of another collision occurring increase slightly.

## Effect of different hash functions