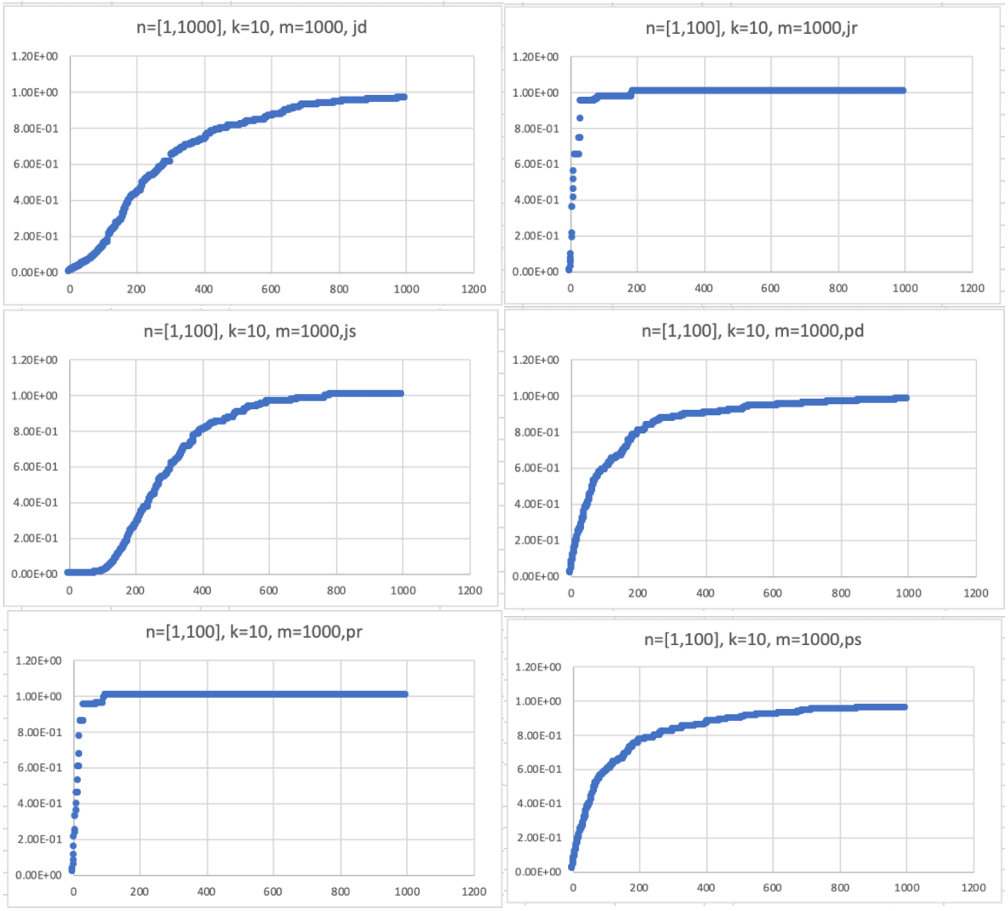
CS130A Bloom Filter Report

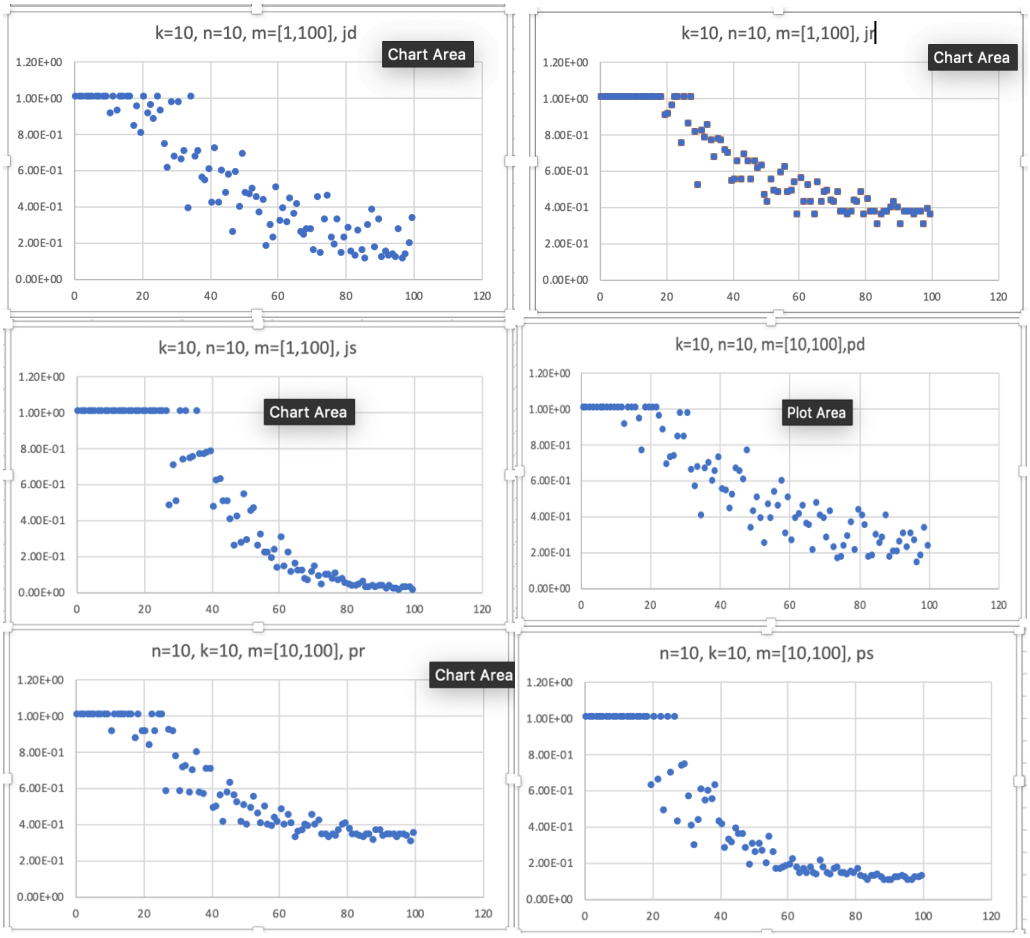
I use the data set of 100000 names and the following are distribution chart and evaluations for hash function pairs.

1. When m = 1000, k = 10 and n ranges from 1 to 1000.
   Distribution charts:



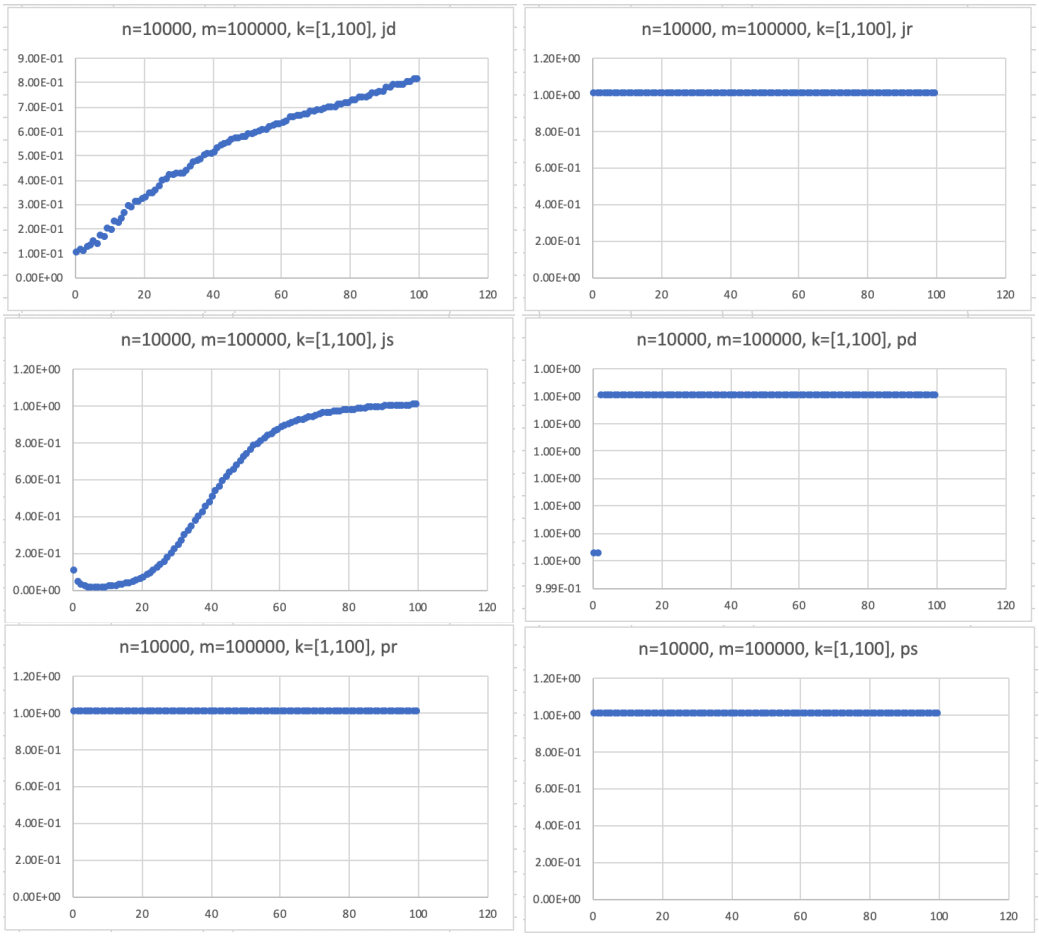
From the graph above, we can conclude that jd and js pairs have best performances when n is increasing. Of all the functions, pr and jr performs worst as n increases because they use reciprocal method. For reciprocal functions, they can produce only a small number of results so they only occupy a small part of the filter. Then as n increases, the filter is only partly occupied and reach a “saturation” point after all the hashing results are achieved and thus it may produce many false positives.

2. When n = 10, k = 10 and m ranges from 1 to 100.
   Distribution charts:



From the charts above, we can see that js pairs and ps pairs are more effective than others and tend to have lower false positive rate and jr and pr are least effective. Jr and pr can hardly achieve 0. This is because they both use reciprocal methods, which is largely dependent on the value of k since each for each i in k, it can only create limited number of results. And since k is a relatively small value, they only small occupy small part of the filter even if m is increasing. Hence, its false positives can hardly reach 0.

3. When n = 10000, m = 100000, and k ranges from 1 to 100.
   Distribution charts:



From the distribution above, we can find that jd and js pairs are more effective than other pairs when k is increasing. Js's performance improves firstly and then get worse. This phenomenon is caused by the reciprocal function. As k increases firstly, we are more confident to compare two strings with more bits available to compare. But as k increases, there are too many occupied positions so it causes more false positives.

Conclusion:
Based on the performances of hash function pairs in all circumstances, I find that Jenkins and Division hash pairs and Jenkins and Squareroot hash pairs are the most effective for bloom filters and they produce the least false positives when number of elements, number of bits and number of hash functions are constants.