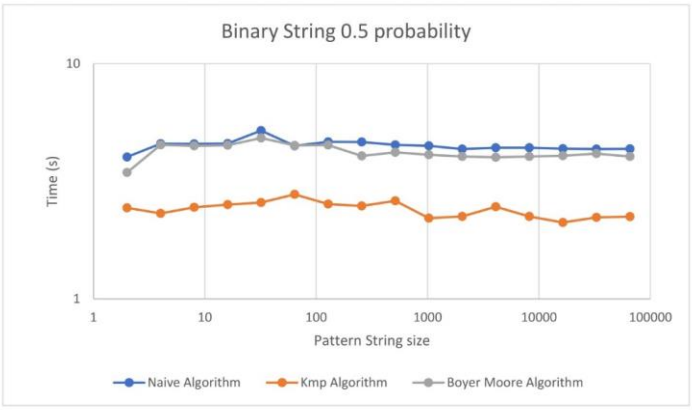
Carson Sorenson

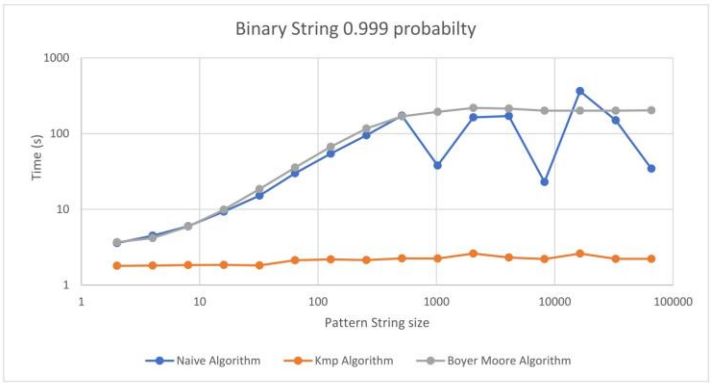
A01988754

CS5050

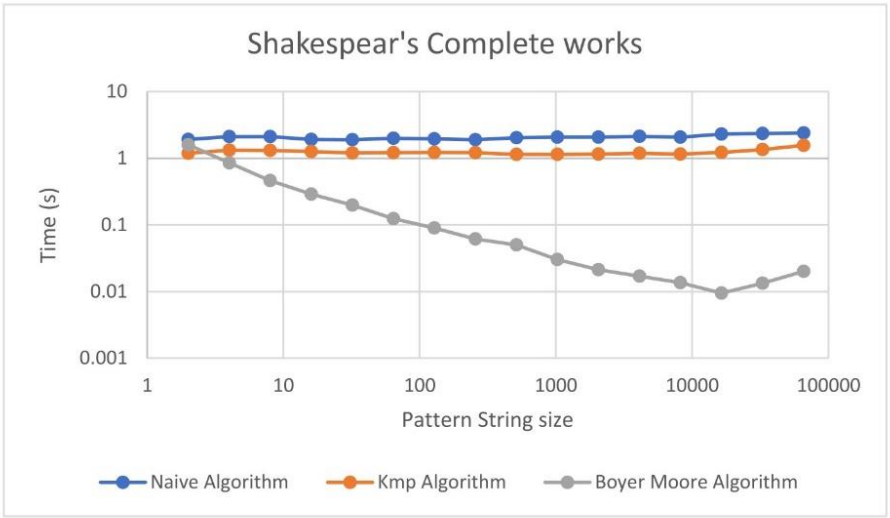
String Matching

The results of the two synthetic binary strings were similar. The KMP algorithm was able to handle substrings smaller than the string size much faster than the BM algorithm and Naïve algorithm because there were only two characters being used. The BM algorithm slows down on small alphabets because substrings re-occur so frequently. This is why the KMP algorithm was so much better on the synthetic binary strings.

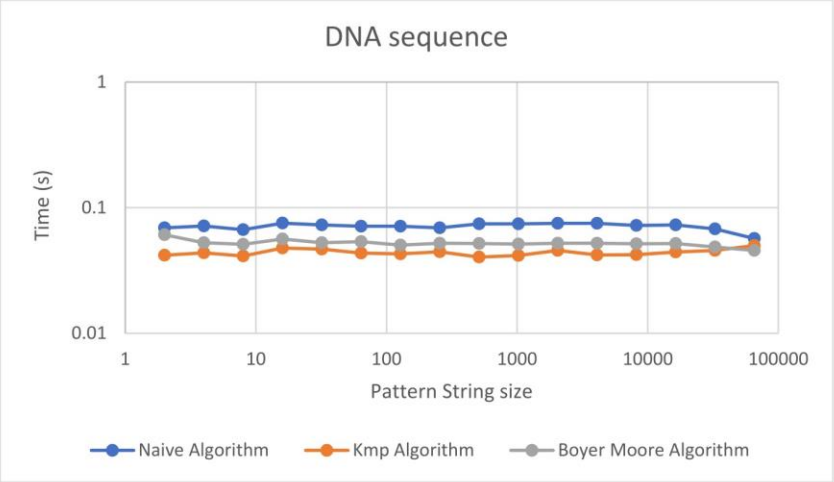




The results of Shakespeare's complete works were very different than the synthetic binary strings. Because the alphabet being used was so much larger the BM algorithm was able to perform much faster than the other two algorithms. The BM algorithm was able to make bigger jumps because of the larger alphabet which meant a faster runtime.



For the DNA sequence the alphabet used wasn't very big so the KMP algorithm was the optimal algorithm. However, all the algorithms were very close in time, but because of the small alphabet in DNA sequences the KMP algorithm was better suited for the search.



In conclusion, with the implementation of the Boyer-Moore algorithm it tends to not perform very well on strings with small alphabets, and especially poor on small alphabets with a lot of repeating characters. Because of this, if I was dealing with a small alphabet, I would choose to use the KMP algorithm to do all string searches. However, in a normal alphabet with a lot of different characters, the Boyer-Moore algorithm was the best, and I would use that in every other situation.