Project 4 Report

I used a hash table as my table data structure. My implementation of the hash table as class HashTable is a dynamically allocated array of STL lists of Sequences. Sequence is a struct I created that contains a sequence of characters as a string and the offset where the sequence was found in the old file as a size\_t. Sequence also has a constructor that initializes its two member variables. HashTable has a constructor that dynamically allocates an array of the size passed in as a parameter. The class also has two member functions for inserting a Sequence and getting an offset from a string passed in as the key if there is a Sequence with the string in the hash table. A private member function hash takes a string as a key and computes an index by using the FNV-1a hash function and then modding the hash by the size of the array. The insert function avoids adding duplicate Sequences by checking if a Sequence with the same sequence of characters already exists before adding the Sequence. HashTable does not have an erase function as it’s not necessary to erase any Sequences when I use it for createDelta. A related non-member function I created is hashSize, which computes the size of a hash table needed to store sequences from a string by taking into account the size of the string, the length of a sequence, and the desired load factor. This function allows createDelta to construct hash tables with the appropriate number of buckets for files of different sizes and sequences of different lengths.