First I transfer the old file into a large string, and for every consecutive 8-bytes string, and the offset where it starts, I store them into a bucket. Then I create an array of buckets, which in total, stores every small 8-bytes strings and their offsets, as a big table. Then I transfer the new file into a large string as well, and for each 8-bytes string, I search them from my table, to find those offsets that satisfy the condition whose string is identical to the string passed in. And I create a vector to store those offsets. For each offset, I find out the length of matching, and by comparison, I find the longest matching length, and use that offset for the copy construction. If I cannot find any string in my table that satisfies the condition, I will add them by writing the adding construction.

So in all, my data structure is an array of buckets plus a vector of ints. It’s a little bit slower than a hashtable, but a lot easier to write as well.