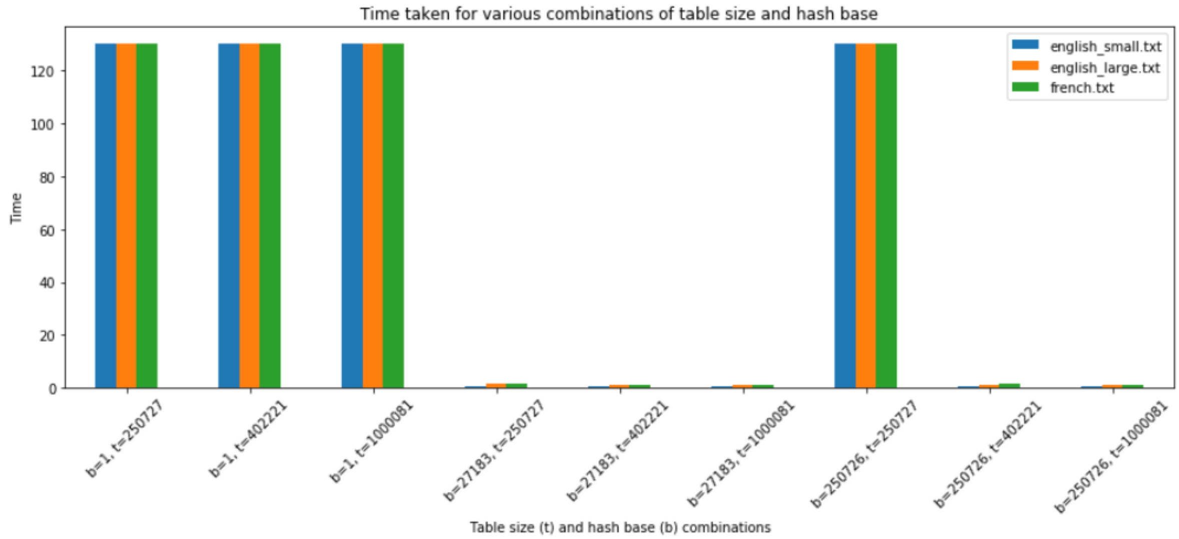
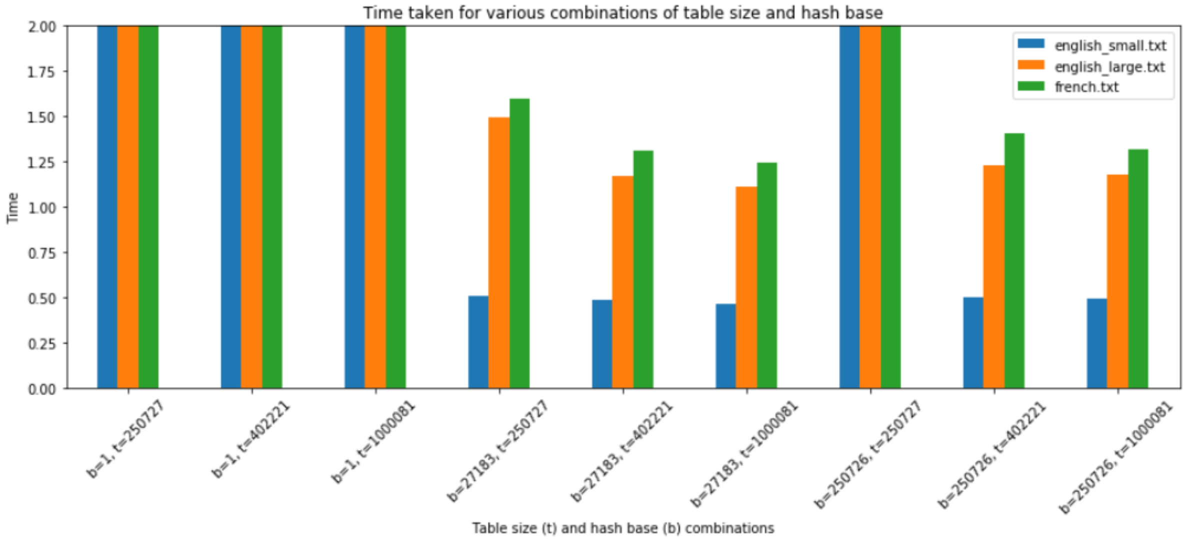
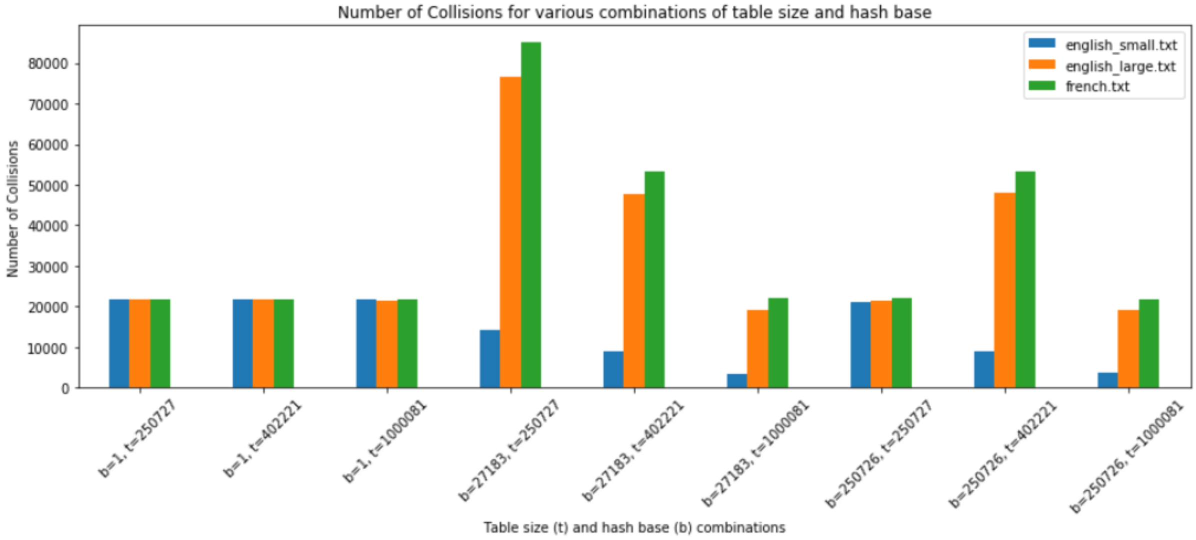
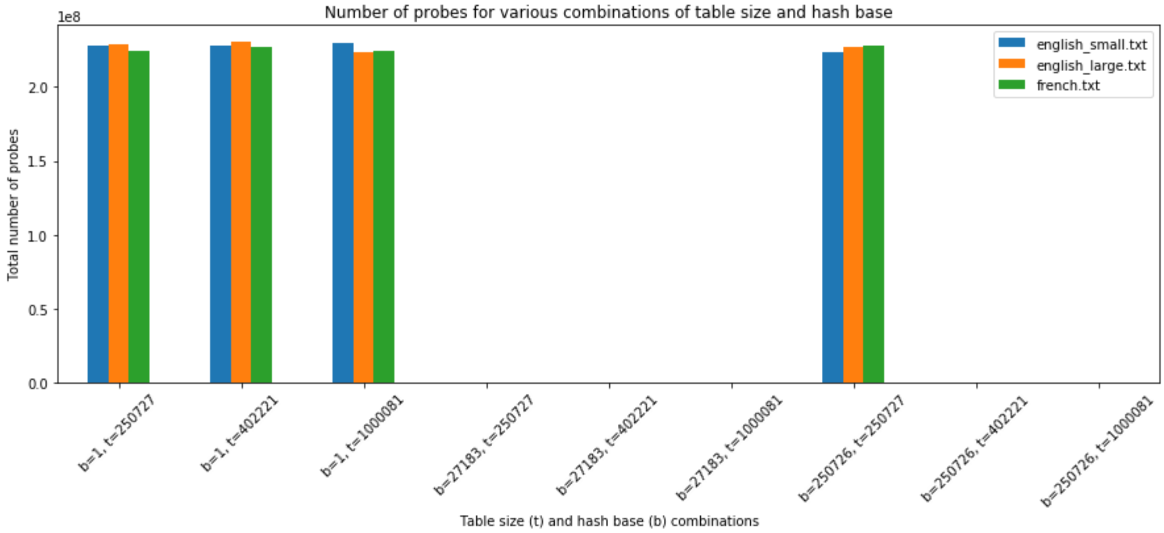
TASK 3



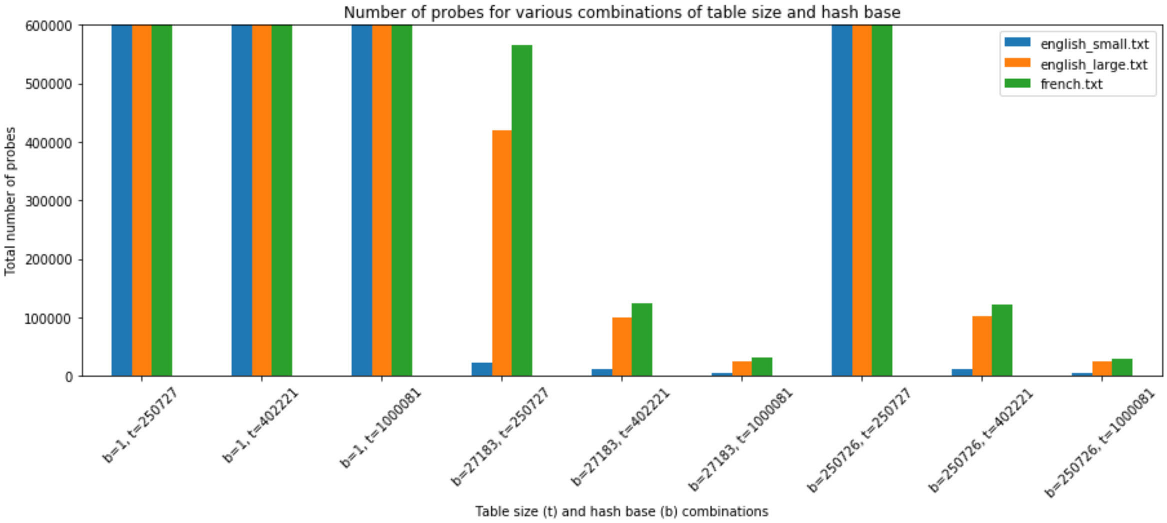
A closer look for runtime (y-axis is from 0 to 2):

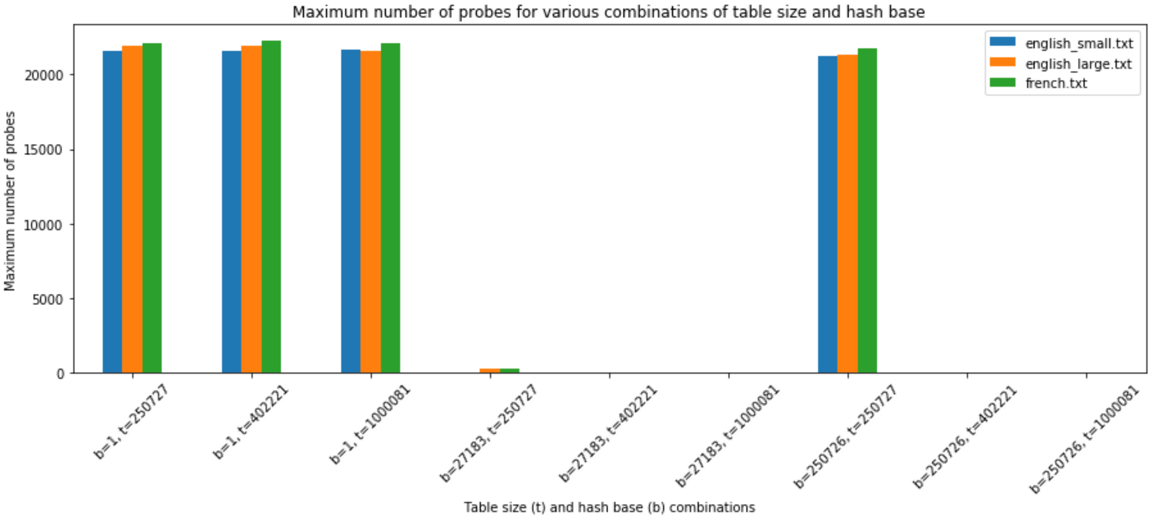




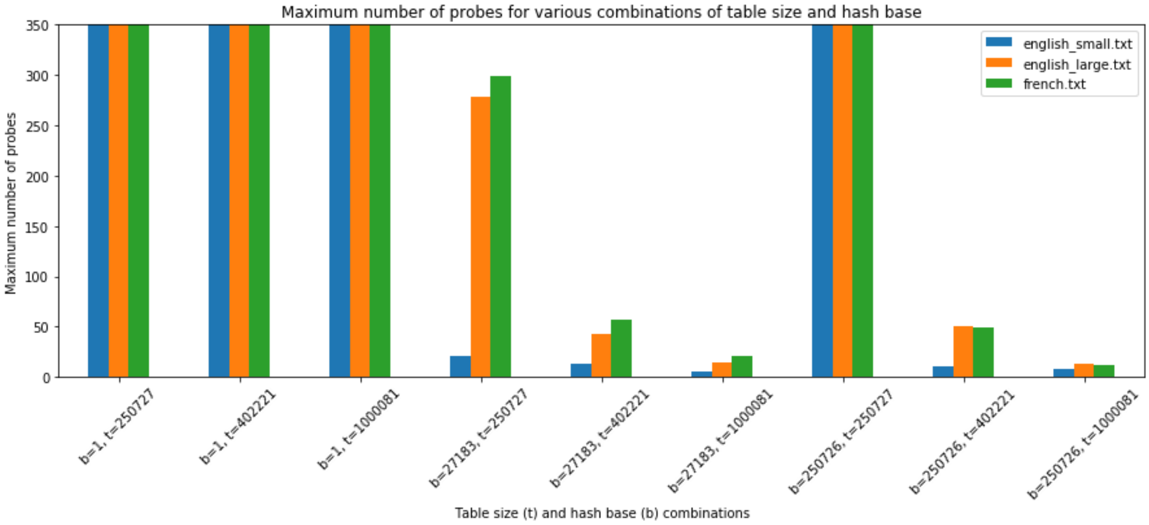


A closer look for total number of probes (y-axis is from 0 to 600 000):





A closer look for maximum number of probes (y-axis is from 0 to 350):



The number of collisions affects the runtime. The higher the number of collisions, the higher the runtime. However, in the cases where the function timed out, the collisions here are low because they probed for a very long time. So, it looks like the number of collisions are low but only because it recorded the number of collisions so far. It also appears that probing affects the runtime a lot because for those combinations which had very high maximum number of probes and total number of probes timed out. When there are less collisions and less probing, the time complexity is closer to O(1) because the key and value can be inserted into the next empty spot. But for the cases where the base is 1 or the combination is base of 250726 and table size is 250727, there’s a lot of collisions and probing and so the time complexity is closer to O(n) where n is the length of the hash table. The rehash_count is 0 for all cases because the table size is big enough that it never becomes full or at least not before the time runs out.