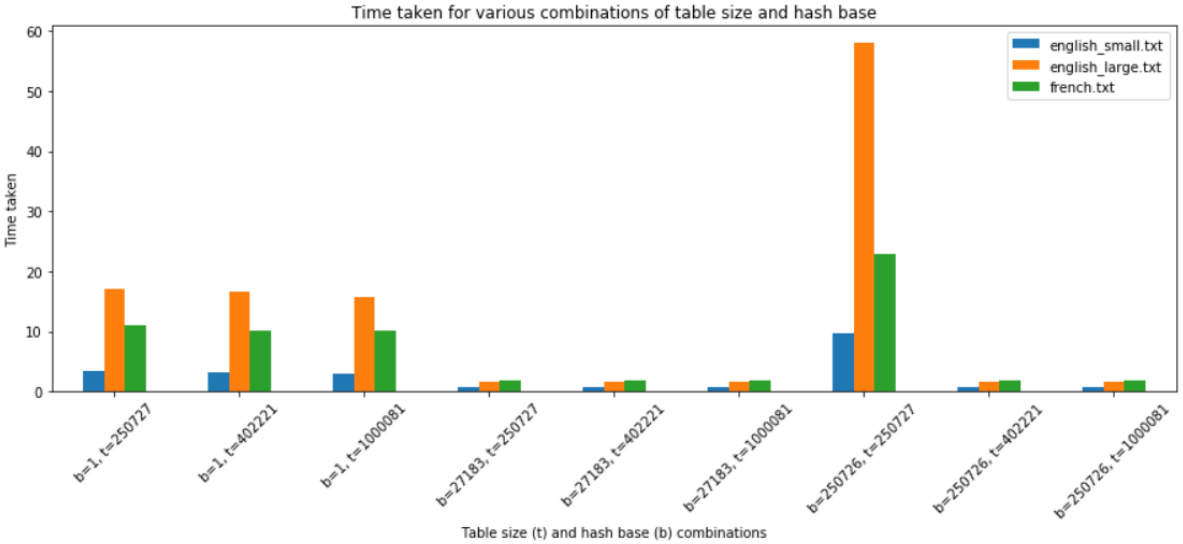
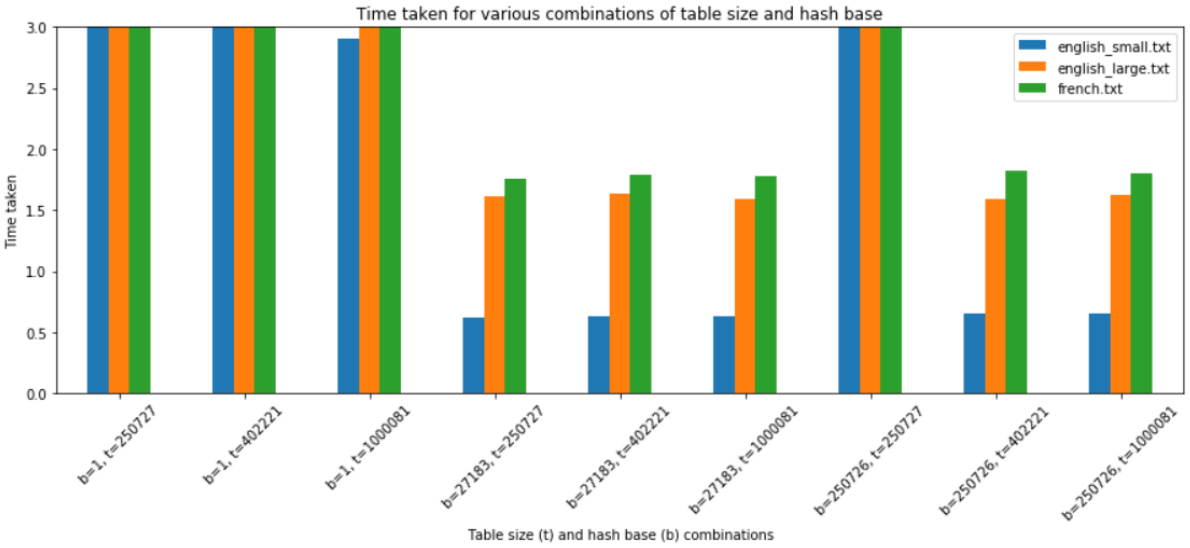
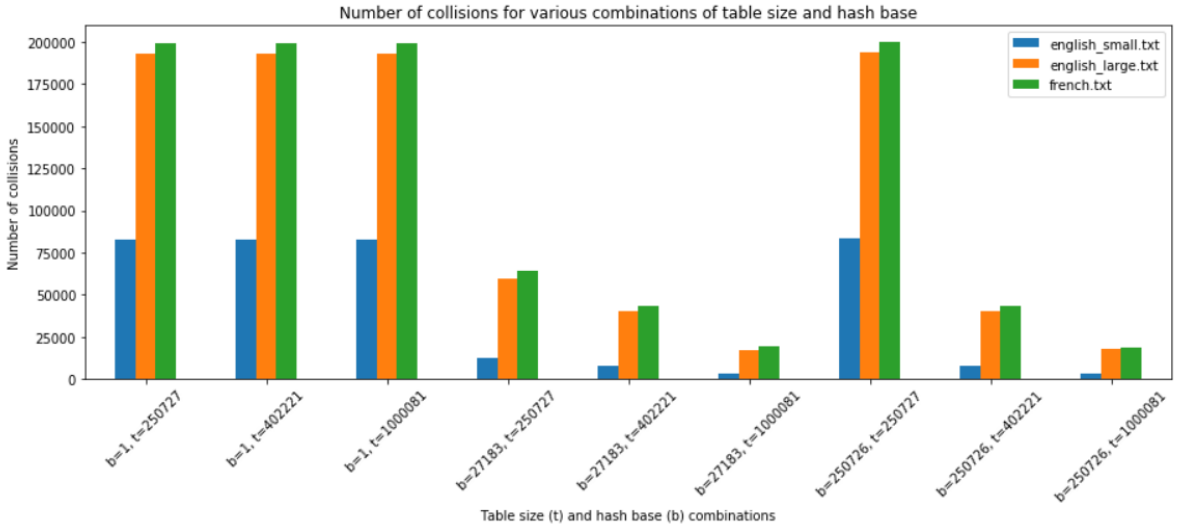
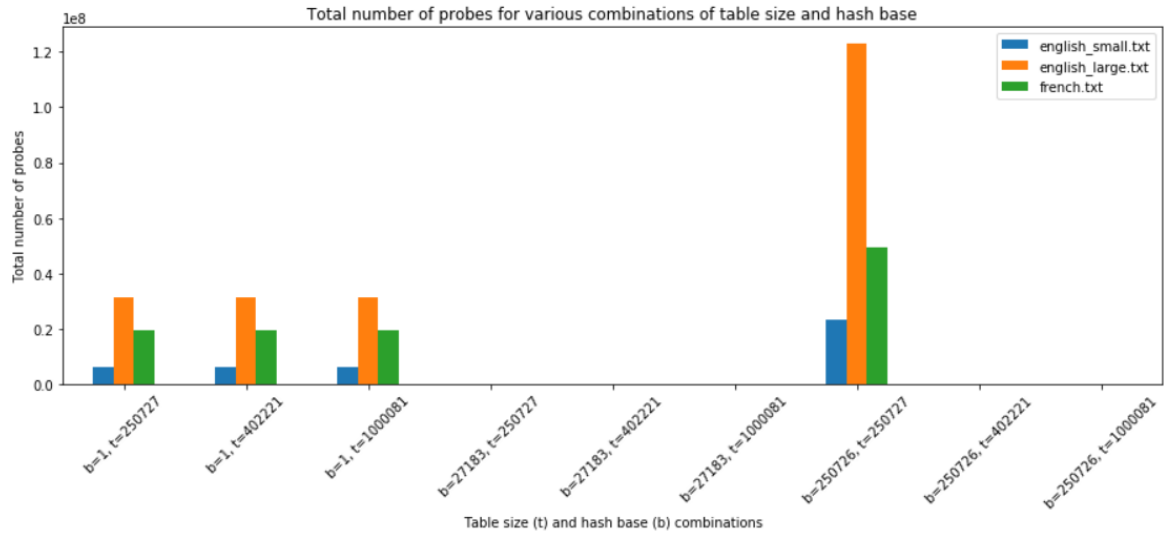
TASK 5



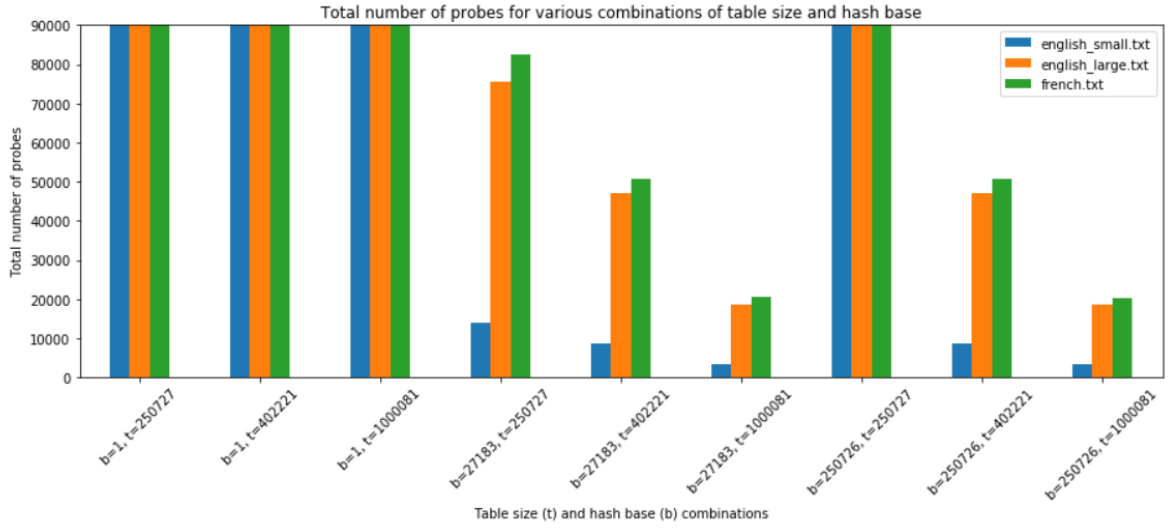
A closer look for runtime (y-axis is from 0 to 3):

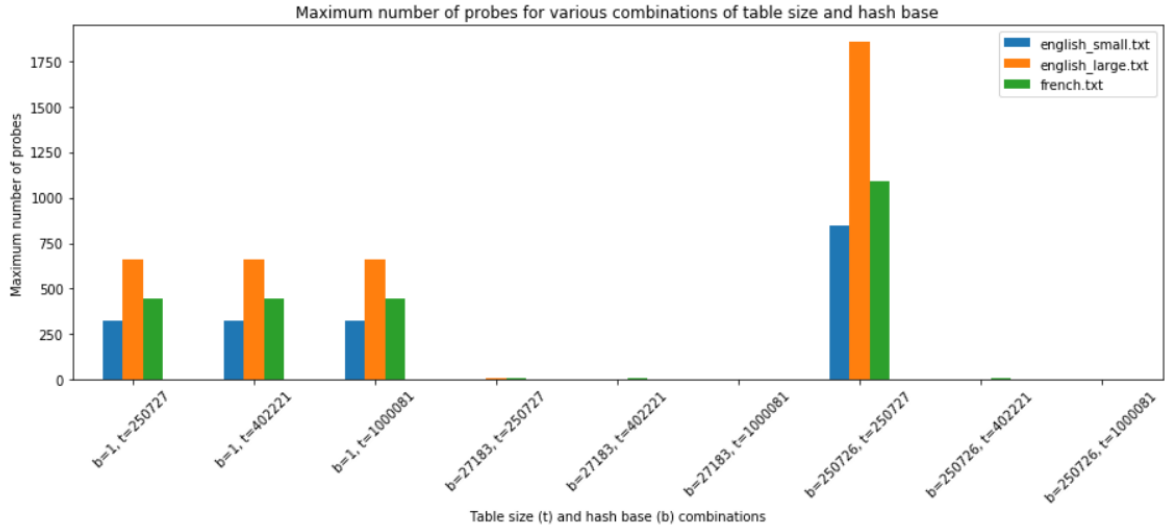




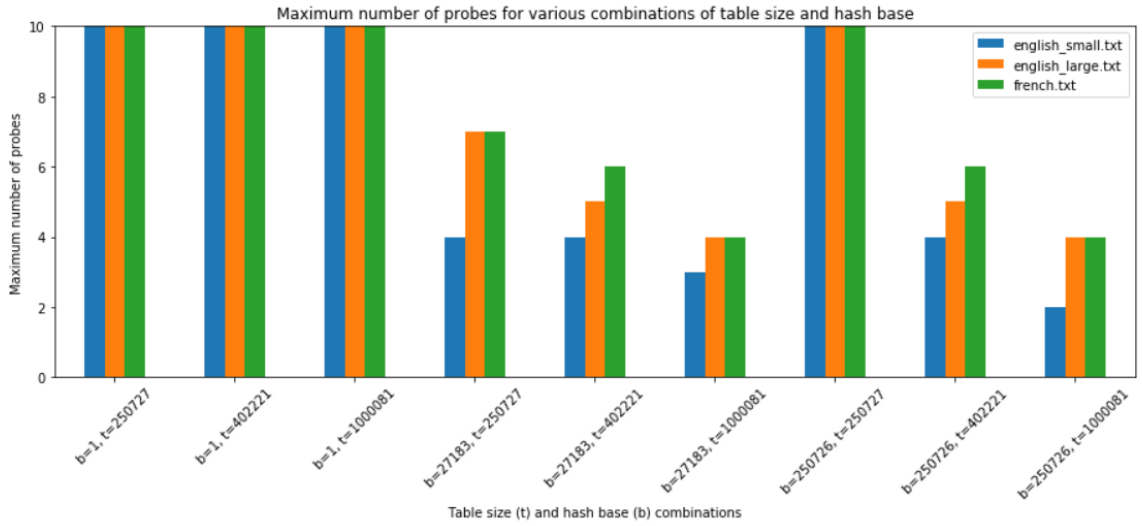


A closer look for total number of probing (y-axis is from 0 to 90 000):





A closer look for Maximum number of probes (y-axis is from 0 to 10):



Separate chaining is a lot faster than quadratic probing or linear probing. It had no time outs and was overall faster compared to the other two previous tasks. Separate chaining still has similar number of collisions to quadratic probing and higher number of collisions to linear probing because collisions in separate chaining happens when the spot in the hash table is taken by a binary tree and this happens quite often. But the maximum and total number of probes in separate chaining is a generally lower than quadratic probing and linear probing because it compares keys and sorts them to the left and right depending if it is smaller or bigger than the key of the root. As with linear and quadratic probing, combinations where the base is 1 and the table size is too close to the base results in higher runtimes. However, it seems that separate chaining is a little less efficient for the dictionary english_large.txt. This could be due to the words in the language itself being too similar which results in unbalanced binary search trees and brings the time complexity closer to O(n) where n is the height of the binary search tree. rehash_count is still 0 because there is no need to call rehash at all since the key and value is just added to the binary search tree in the hash table.