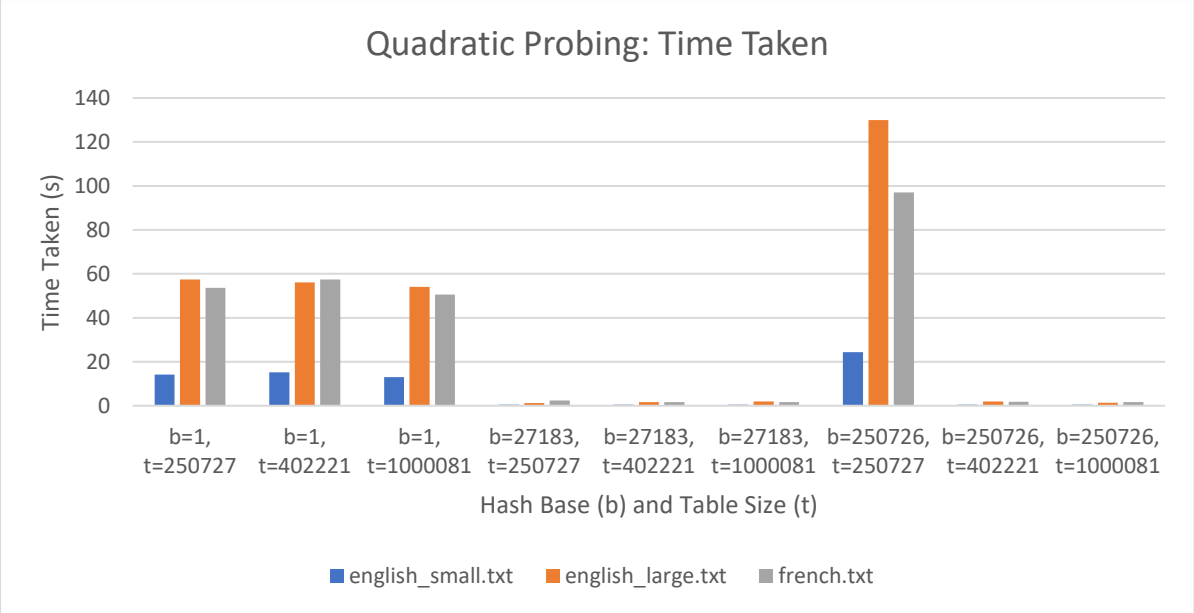
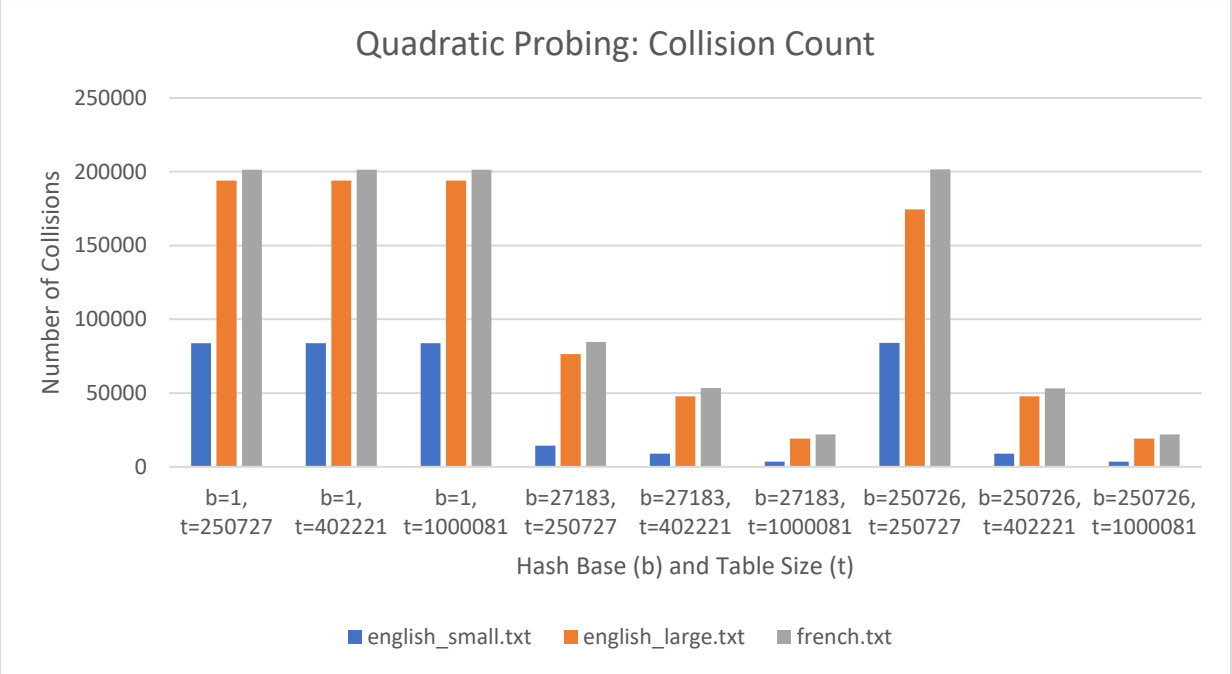
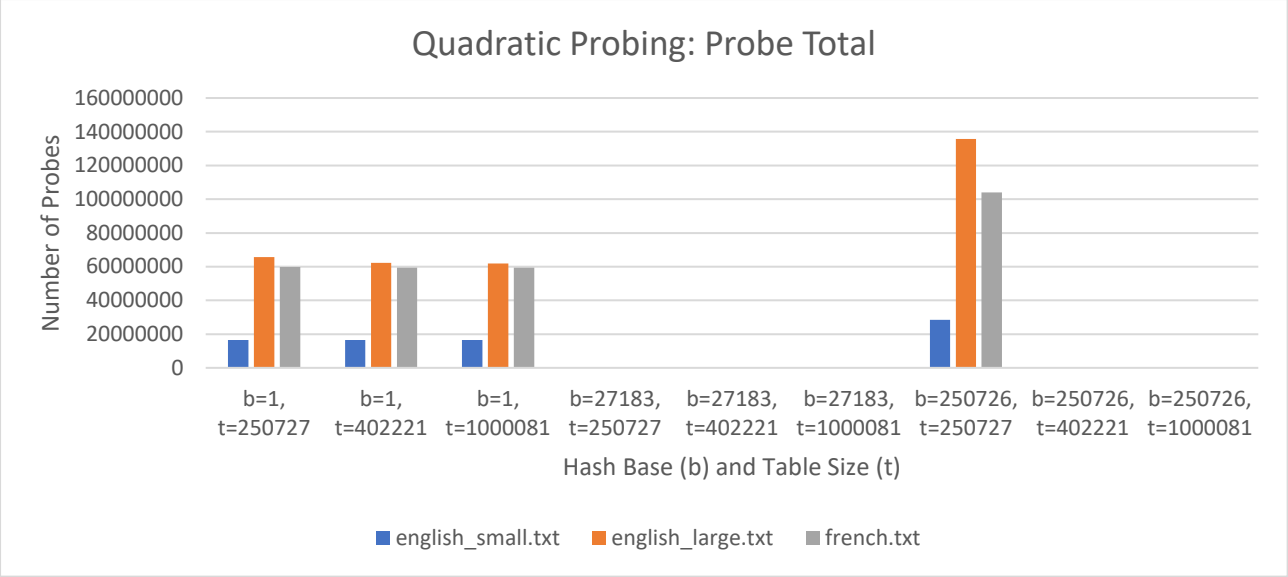


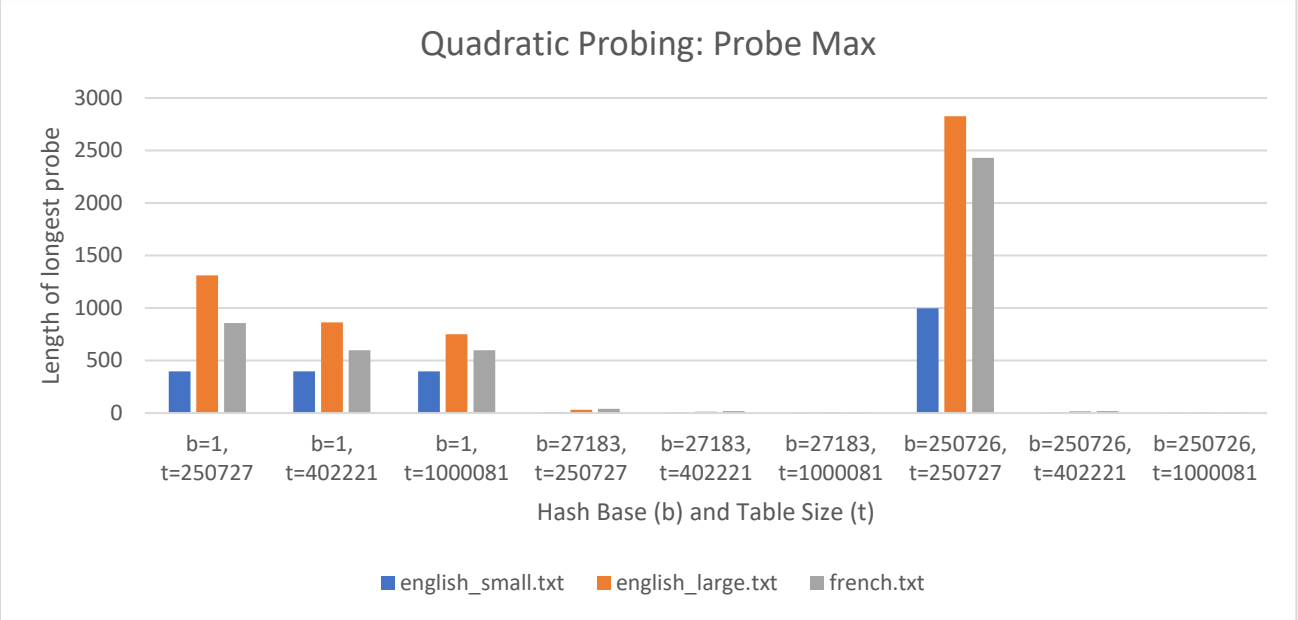
In the graph above, it can be seen that no hash base and table size combination timed out for the dictionaries (with the exception of “english_large.txt with a b=250726 and t=250727). This is as a result of using quadratic probing instead of linear probing, which decreases the number of times that the list needs to probed as it eliminates the problem where two different keys that hash to different values are in the same cluster (primary clustering).



In the collision count graph shown above, it can be seen that there are a higher number of collisions when compared to linear probing for the first three combinations. As quadratic probing doesn't insert the next item into the next position when there is a collision, there is spaces between items where an item could be stored. Collisions are more common however, as the hashed items are spread out throughout the table.



In the graph above we can see that between linear and quadratic probing, quadratic probing has the lowest probe total between the two methods. This is because that as a result of using a quadratic probing method, only keys that hash to same hash value will result in a probe. This results in a smaller amount of probe chains.



In the graph above, we can see that the probe max count of quadratic probing is much lower than linear probing. This is because it eliminates the problem of two different keys that hash to different values are in the same clusters. Probing only happens now when the keys have the same hash value, which results in much shorter probe chains.