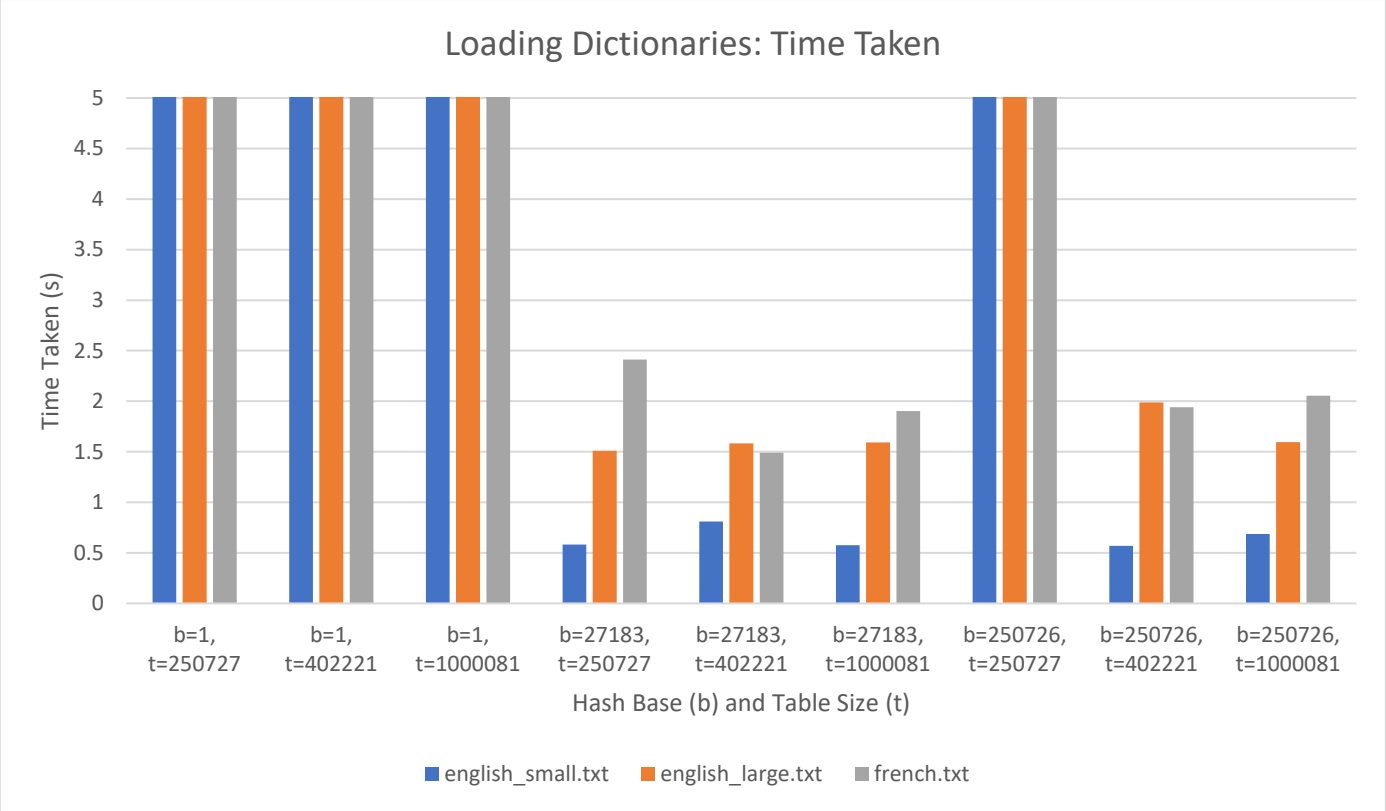

In the graph above, we can see the time taken for each dictionary to be hashed into a hash table of varying sizes and using different bases for the hash function.

With a hash base of 1 and any table sizes, it can be seen that loading the dictionaries takes too long and the program times out. This can be explained by the small hash base causing the hash function to create the same hash value for many keys. This would then require the program to probe multiple times in order to find an empty space to insert the item.

Another hash base and table size pair that works badly is b=250726 and t=250727. This could possibly be explained by how close the two values are, which may result in the hash function giving out very similar values.

For the rest of the hash base and table size combinations, “english_small.txt” typically finished first as there were much less words in the dictionary to input and thus, less chance of collisions. “english_large.txt” and “french.txt” usually took longer, because of the increased number of words in the dictionaries. This meant that as more values were hashed into the table, the chances of a collision increased and thereby increasing the time taken for the program to finish as it is required to prove for empty spaces to insert the item.