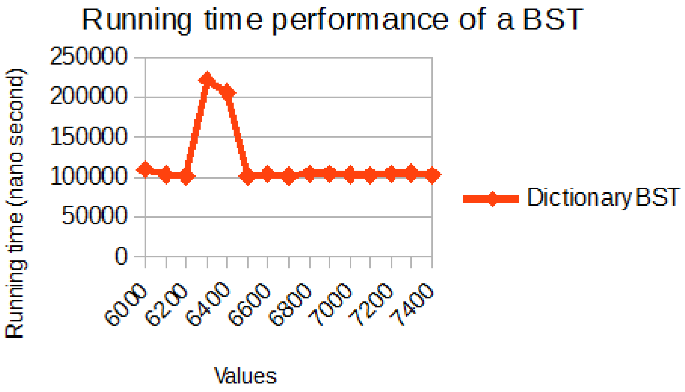
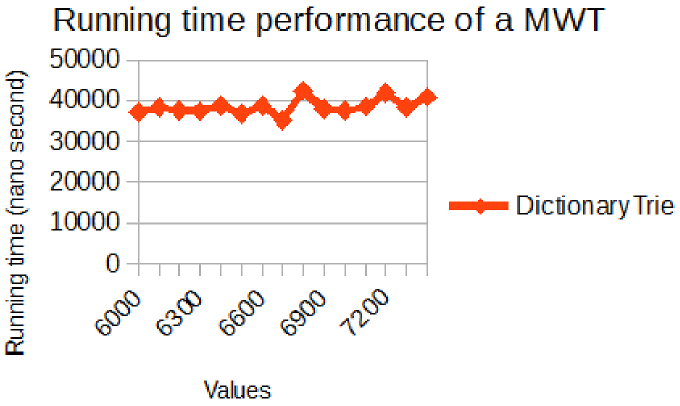
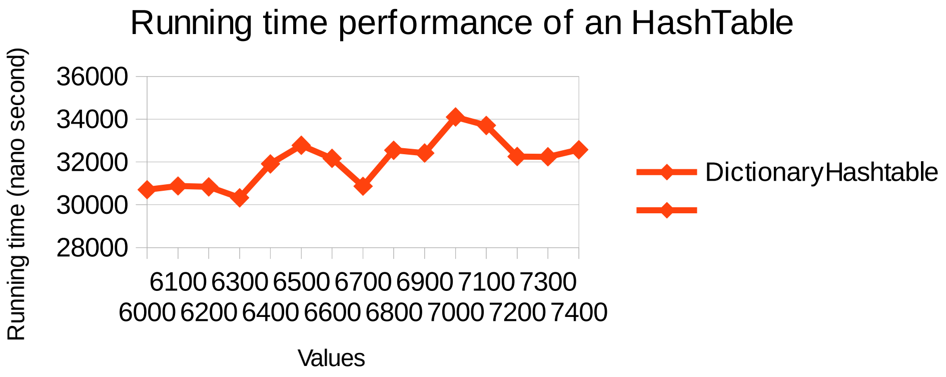
Martin
Nicolas
A14283795

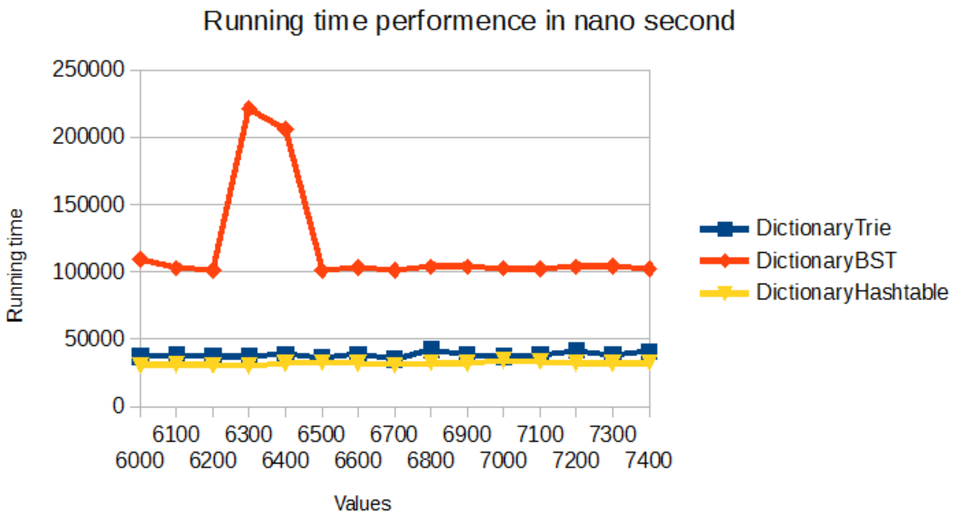
Sorry for my english : i am an international student from France

1.1 The results makes sense with these analytical running time expectations. First of all, we can see that for the HashTable the average is about 32 000 nano seconds, for the MWT is 40 000 ns and for the BST is 102 000 ns. The complexity of an Hashtable is O(1) it is the lowest average case time to find, indeed we can see that is the lowest with 32 000 ns. The BST has the biggest average case time here to find it is O(log(N)) so it has to be the biggest value and it is with 102 000 ns. The MWT has an average case time to find of O(N) so more than an HashTable but a lot less than a BST ,the results corresponds with 40 000 ns.



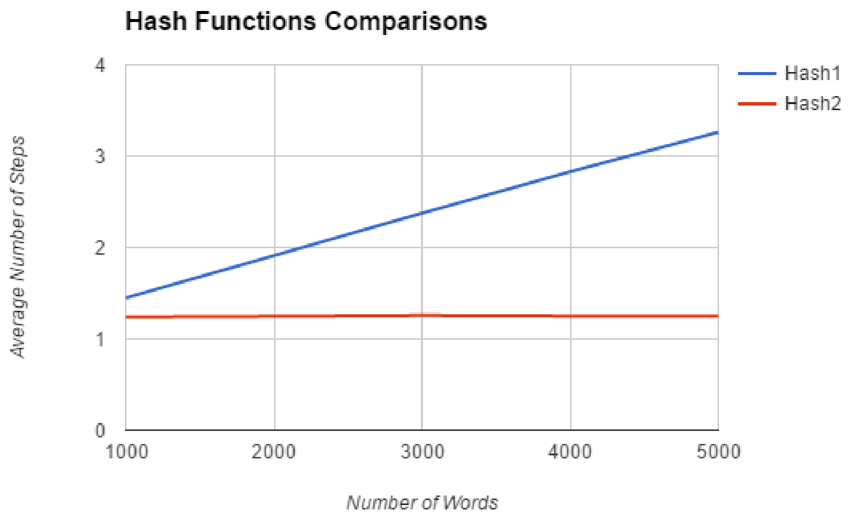






The hashFunction1 works adding the ASCII values of each character of the word to a variable and at the end dividing it but the size of the tableSize (source : stackflow) . The hashFunction2 it multiplies the seed by the current accumulation of characters at thatr point and keep adding it together and then divide it. (stackflow)

1. hello : 532 hi : 209 then : 431
2. hello : 622 hi : 1729 then : 641



The second hash function is better because the average number of step stay constant as the number of words increases trend it stay over one step which is very good compared to the other hash function which increases over time and approaches average of 3 steps.