Experiment Inserting Performance

implementation of insert for hash sets

Hypothesis

The Hypothesis is the expected complexity of the hash-set

Theoretically, the average case for insertion operation is O(n) if the query file is empty when the input is randomly chosen, and the expected complexity of the single insert operation is O(1)

The running time of the whole programme increases linearly when the dictionary size increases, but the average running time of the insert operation is constant.

As the programme consists a sequence of insertions, if the expected behavior of the programme is O(n), then the expected average performance of the insert operation is O(1)

Here are some assumptions for this average case assumption:

All operations are constant.

The string is transferred to a hash key and it is inserted into a Hash Set based on the key, if the certain slot is already occupied, then it will check the next slot until it finds an empty slot

Design

The input files

To test the hypothesis, 7 random input dictionaries of sizes 100K, 200K, 300K, 400K, 500K were produced. In order to reduce the probability that some random generated dictionaries have non-typical property of the average case, This experiment generates 5 for each so it can get an average running time for each size.

The process of generating dictionaries input was automated using the shell scripts shown in Appendix A. The script for computing the time shows in Appendix B.

The time for the program to execute was measured using the UNIX time command summing the user and sys values output by the command.

Results

The results were then plotted using matplotlib and excel’s fit functionality was used to fit a line f(x) = m × x + q calculating values for m and q in the process. The results are shown in Figure 1 and the raw data can be seen in Appendix C. As can be seen from the graph the the line f(x) = mx+q has a good fit and the values of 0.000024546 and 0.0104 have been computed for m and q respectively. This confirms the hypothesis about the behaviour of the algorithm on average input, and allows us to predict the time taken to sort a dictionary of size x as 0.000024546 + 0.050390325.

Hypothesis

The Hypothesis is the expected complexity of the BSTree

The behaviour for insertion on binary tree is somewhere between the theoretical average case O(log(n)) and the theoretical worst case O(n).

To test the hypothesis, 7 random input dictionaries of sizes 100K, 200K, 300K, 400K, 500K were produced. In order to reduce the probability that some random generated dictionaries have non-typical property of the average case, This experiment generates 5 for each so it can get an average running time for each size.

The process of generating dictionaries input was automated using the shell scripts shown in Appendix A. The script for computing the time shows in Appendix C.