Comparing Dictionary Structures

Hypothesis

Here I compare two backend implementations of a Dictionary namely Hash Tables and Binary Trees. Now, we have learnt that the insertion time for a dictionary based on Hash Tables is O(1) whereas the insertion time for a dictionary based on Binary Trees is $O(\log_2(n))$. Now, we obviously know that Hash Table implementations are much faster than Binary Trees but guessing the number of insertions for which one implementation will take over 3 sec will depend on the CPU clock speed and therefore is hard to do. I would therefore make a random guess as 10^6.

Methods

The default available implementation of a dictionary in python uses a hash table as the backend and therefore for the hash table implementation I basically use the inbuilt dictionary available in python. For the Binary Tree implementation no standard inbuilt data structure exists, and therefore I use the implementation available at https://pythonhosted.org/BTrees/ which can be installed in python using **pip3 install BTrees==4.7.1**

We start off with inserting 1 element and increase the number of insertions at each step by a factor of 10.

I compute the time taken by both the implementations to insert the specified number of elements in the dictionary

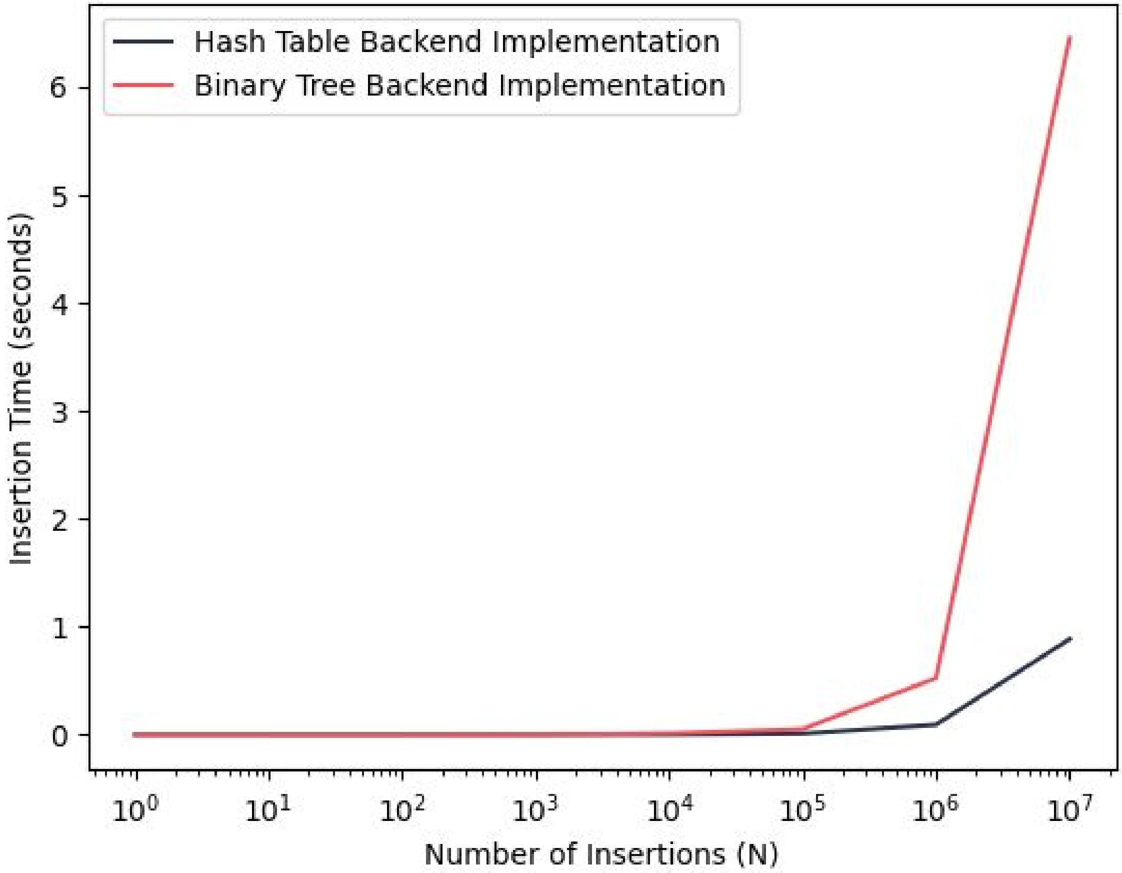
I repeat this procedure until one of the implementations takes more than 3 sec to perform all the insertions.

Finally, I plot the results using the matplotlib library of python.

The code is available at
https://github.com/divyanshaggarwal/CSE830/blob/main/HW4/Ques3/HW4_ques3.py

Results

The following plot represents how the runtime of two different implementations of a dictionary namely Hash Table and Binary Trees varies with the number of insertions being made. The x-axis represents the number of insertions being made on a log scale and the y-axis represents the total time elapsed to make these insertions. The red line indicates the Binary Tree based backend implementation and the blue line indicates the Hash Table based backend implementation.



Discussion

It is quite clear from the graph above that the hash table based implementation of a dictionary is much faster than the Binary Tree based implementation. However, it is interesting to see that for insertions over 10^6, there is a huge difference between the run times of both these implementations. While the hash table based dictionary is able to make 10^7 insertions in nearly 1 sec, the binary tree based implementation takes over 6 sec to make these insertions. Also, it is interesting to see that we do not notice much of a difference in the run time of both these algorithms when we talk about insertions within the range of $10^{5.}$ Therefore, for large values of N, hash table based implementation is much faster for insertions than the binary tree based implementation.

Conclusion

Therefore, we conclude that for large values of N (number of insertions), the hash table based implementation of a dictionary is much faster than the binary tree based implementation.