

Hypothesis. I expect the hashtable to be a lot faster than the binary tree.

Methods. The experiment is ran on Ubuntu 19.04. GCC version is 8.3 and python version is 3.7.3. Also, matplotlib, pandas, and numpy are dependencies of plot.py

To run the experiment, first compile

```
g++ -fconcepts -std=c++2a -O3 code.cpp
```

Then execute

```
./a.out
```

which may take a few minutes to run. To plot the results use

```
python3 plot.py
```

The program compares binary tree and hash table insertion performance for various input sizes n in the range $[0, 4466816]$. For each input size n , the program inserts n uniform random integers into a binary tree and into a hash table. The total time taken for inserting n elements into the binary tree is recorded and similarly for the hashtable.

Results.

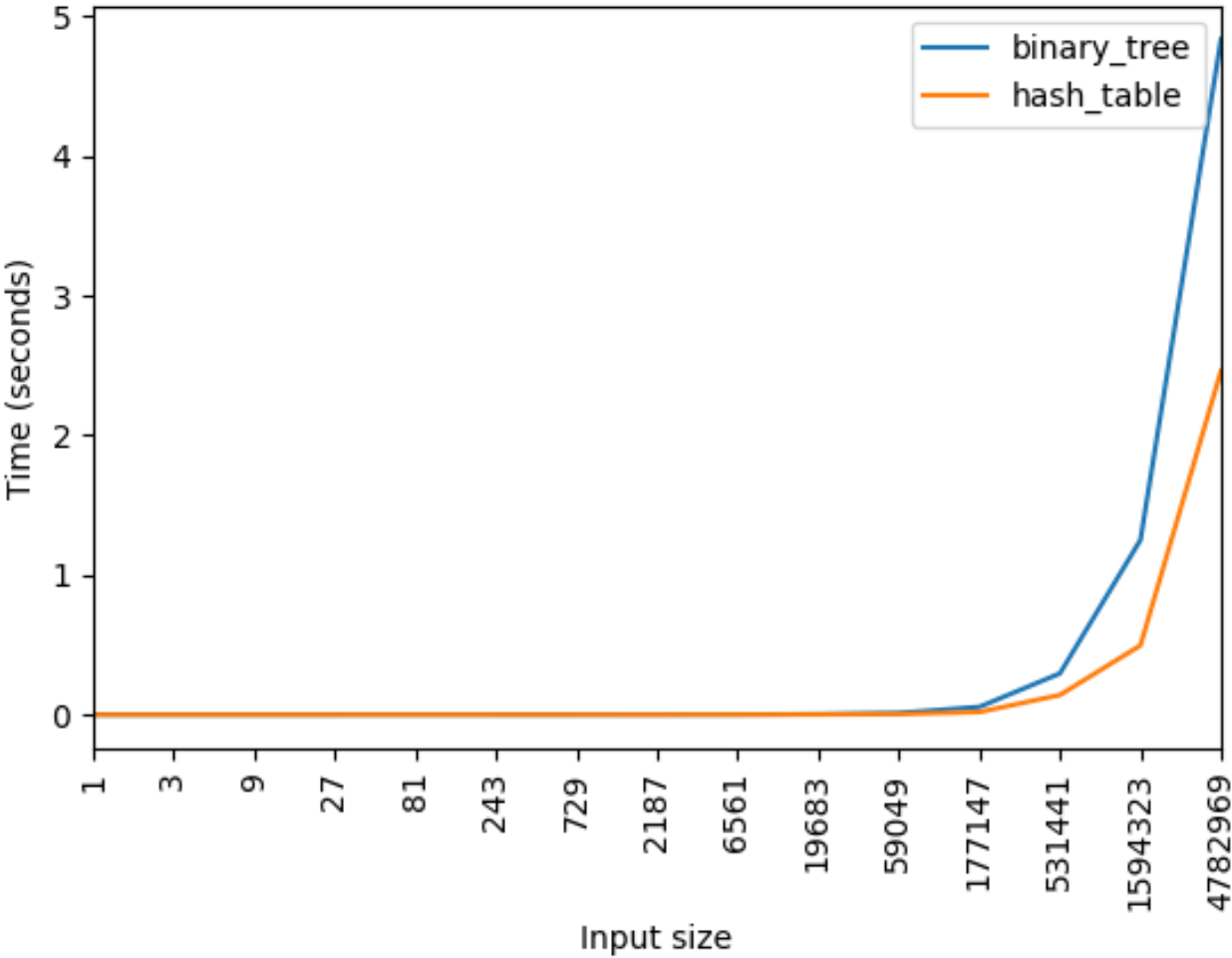


Figure 1

Figure 1 simply gives the time to insert n items vs the value for n .

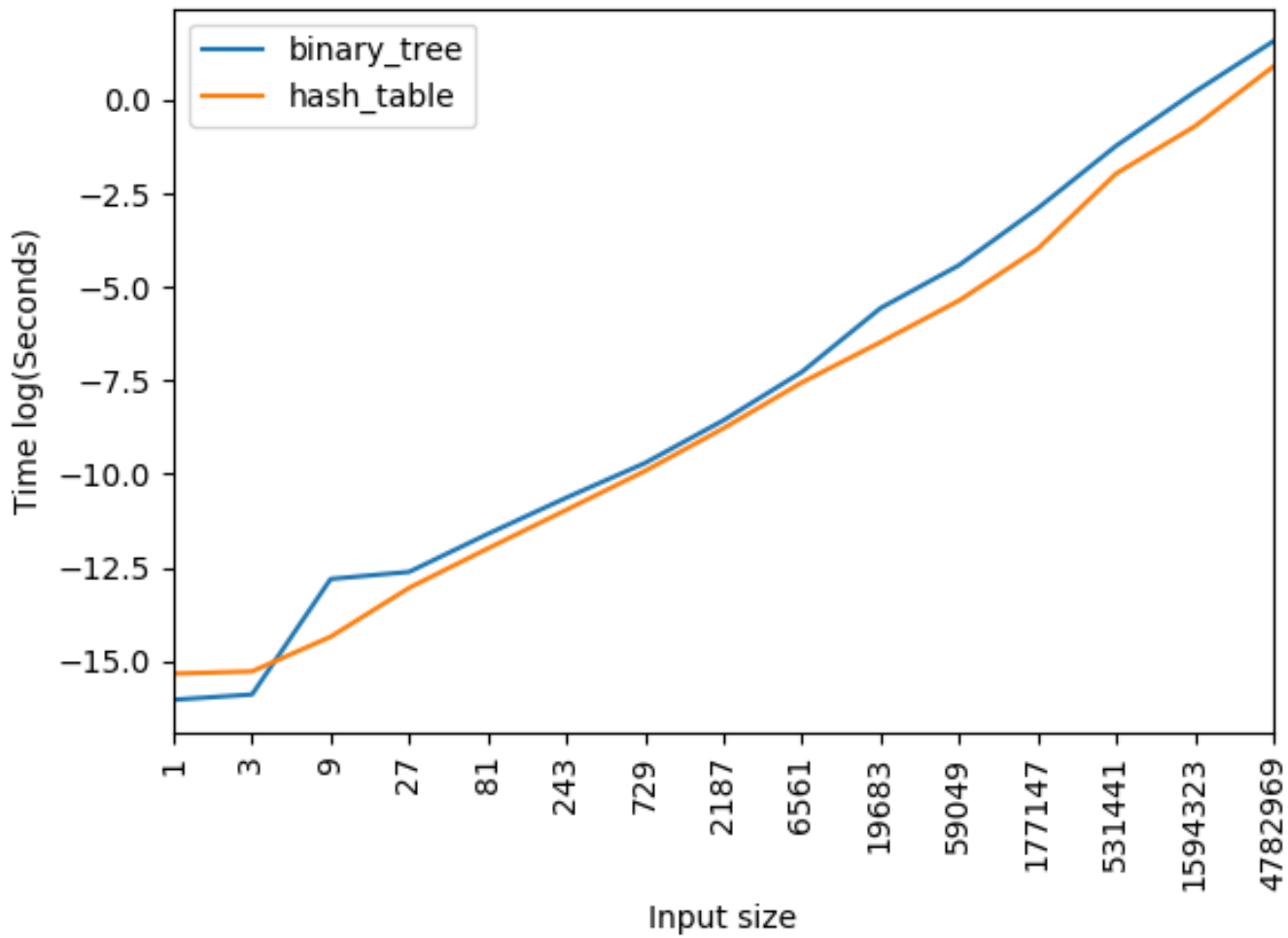


Figure 2

Figure 2 shows the same data as in Figure 1 but with a log scale on the vertical axis.

Discussion. From the plots we can see that the binary tree is competitive with the hash table for input sizes of 10000 or less.

Conclusion. Binary trees are competitive with hash tables for input sizes of 10000 or less.