CEREN ANIL 24108

CS300 HW3 REPORT

Console output of my program;

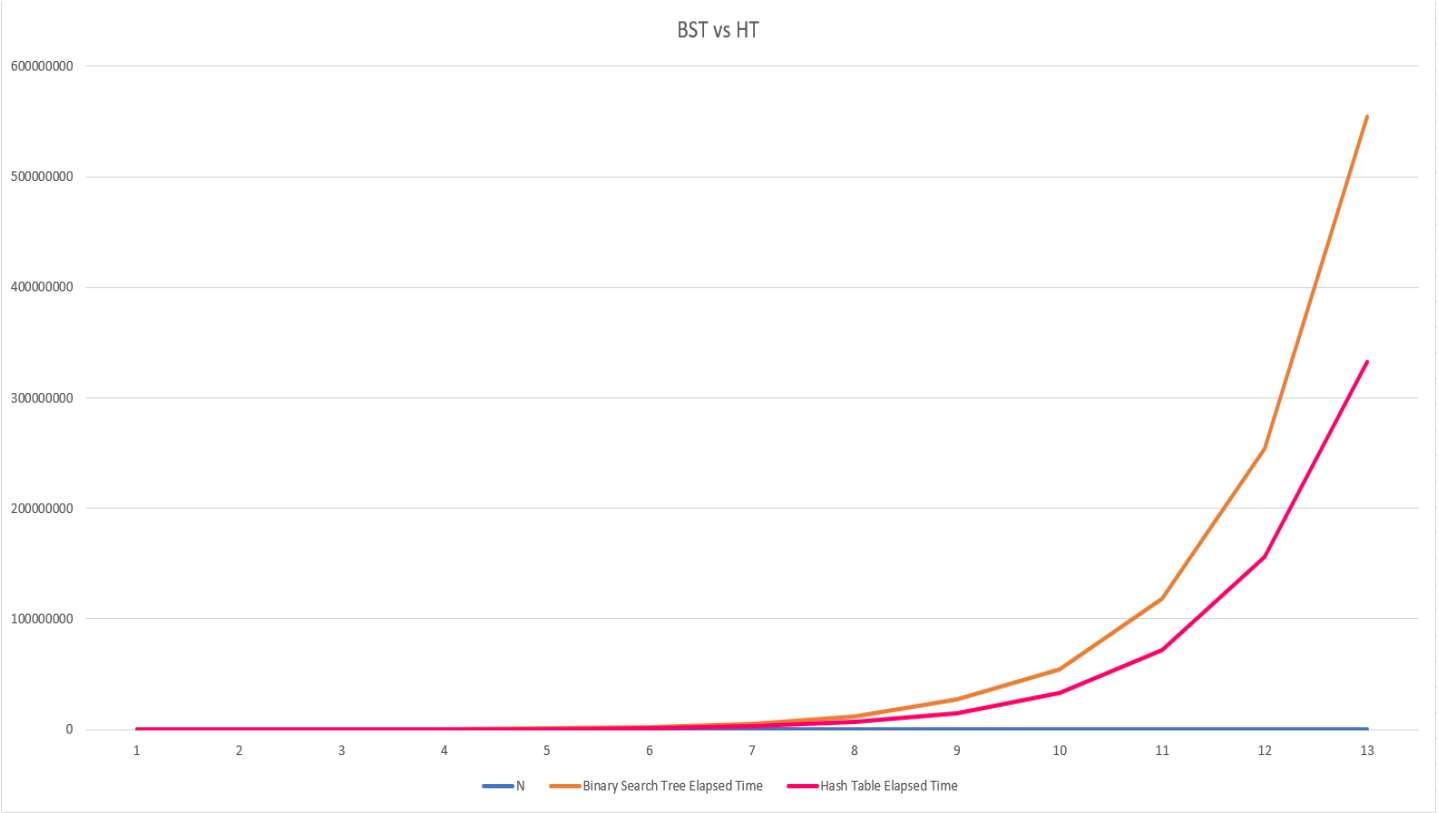
```
Building a binary tree for dict.txt...
Building a hash table for dict.txt...
rehashed...
previous table size: 53, new table size: 107, current unique word count: 26, current load factor: <0.242991>
rehashed...
previous table size: 107, new table size: 223, current unique word count: 53, current load factor: <0.237668>
rehashed...
previous table size: 223, new table size: 449, current unique word count: 111, current load factor: <0.247216>
rehashed...
previous table size: 449, new table size: 907, current unique word count: 224, current load factor: <0.246968>
rehashed...
previous table size: 907, new table size: 1823, current unique word count: 453, current load factor: <0.248491>
rehashed...
previous table size: 1823, new table size: 3659, current unique word count: 911, current load factor: <0.248975>
After preprocessing, the unique word count is 995. Current load ratio is 0.271932
Running queries in query1.txt...

***********************************************
Benchmark results for Binary Search Tree (BST):
***********************************************
+ Elapsed time: 32000 ns
+ Average query time: 32.1608 ns

***********************************************
Benchmark results for Hash Table:
***********************************************
+ Elapsed time: 7000 ns
+ Average query time: 7.03518 ns
+ Speed up: 4.57143 x
```

```
Time measurements in ns (N, 4096N):
*******************************
bst
N                       time
1                       18000
2                       58000
4                       165000
8                       532000
16                      1.202e+06
32                      2.609e+06
64                      4.991e+06
128                     1.1628e+07
256                     2.7288e+07
512                     5.4842e+07
1024                    1.1882e+08
2048                    2.54173e+08
[4096                   5.53896e+08
ht
N                       time
1                       12000
2                       29000
[4                      107000
8                       242000
16                      550000
32                      1.296e+06
64                      2.936e+06
128                     6.647e+06
256                     1.5005e+07
512                     3.332e+07
1024                    7.2137e+07
2048                    1.5605e+08
4096                    3.32719e+08
```

CEREN ANIL 24108

CS300 HW3 REPORT

Graph of my program (Elapsed time(nanosecond) vs N);



As it can be seen from the console output of my program, my Hash Table works 4.5x faster than my Binary Search Tree.