GERALD WORKMAN | TREE | COMPARISONS MADE = 16
JAMES TAYLOR | TREE | COMPARISONS MADE = 12
JEFFERY DORSEY | TREE | COMPARISONS MADE = 16

GERALD WORKMAN | TABLE(1000) | COMPARISONS MADE = 1001
JAMES TAYLOR | TABLE(1000) | COMPARISONS MADE = 1001
JEFFERY DORSEY | TABLE(1000) | COMPARISONS MADE = 1001

GERALD WORKMAN | TABLE(20000) | COMPARISONS MADE = 20001
JAMES TAYLOR | TABLE(20000) | COMPARISONS MADE = 9185
JEFFERY DORSEY | TABLE(20000) | COMPARISONS MADE = 20001

GERALD WORKMAN | TABLE(100000) | COMPARISONS MADE = 4820
JAMES TAYLOR | TABLE(100000) | COMPARISONS MADE = 9185
JEFFERY DORSEY | TABLE(100000) | COMPARISONS MADE = 100001

GERALD WORKMAN | TABLE(100003) | COMPARISONS MADE = 54807
JAMES TAYLOR | TABLE(100003) | COMPARISONS MADE = 65994
JEFFERY DORSEY | TABLE(100003) | COMPARISONS MADE = 100004

Overall, the binary tree had less search comparisons than the various hash tables. This is because the binary tree can implement a form of binary search whereas the hash tables are using a form of linear search. As such, none of the hash tables exhibited better performance than the binary tree. The larger the number of buckets in the hash table, the worse the efficiency.

100,003 buckets may be used if we know the exact size of the data set, in which there would be no point in making a larger table that occupies more memory.