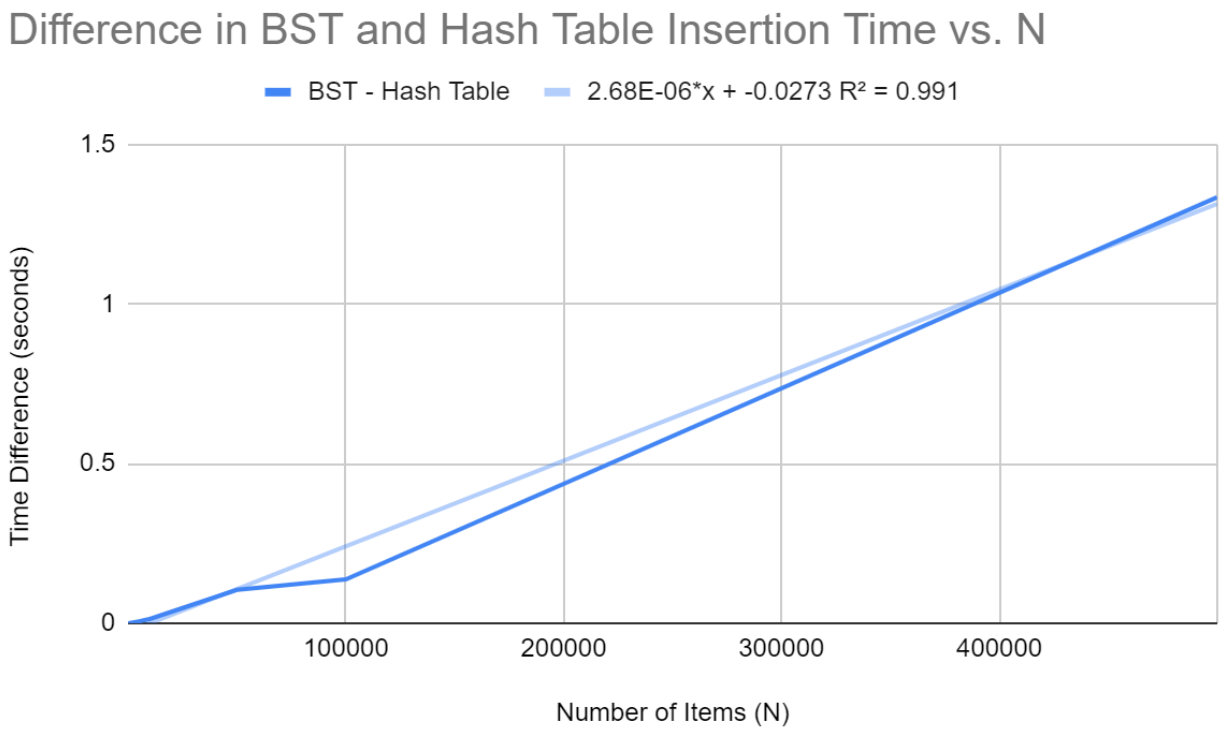
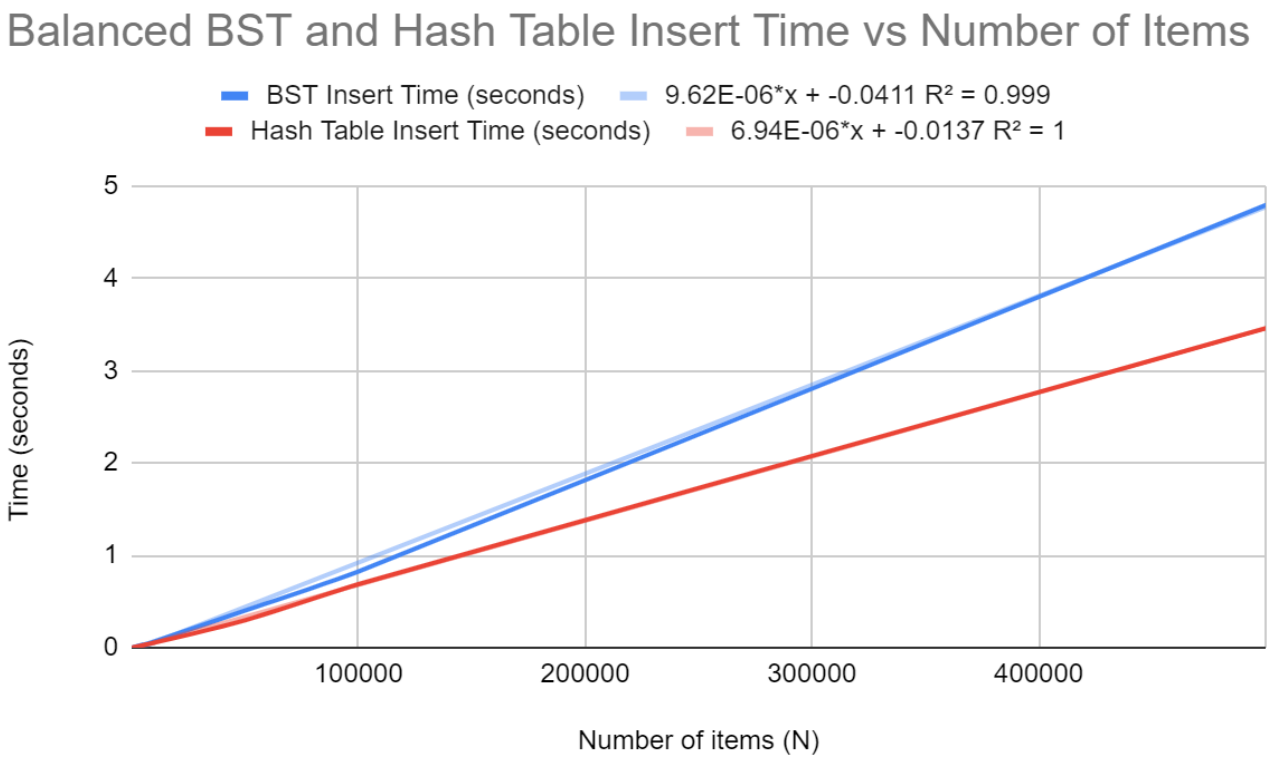
# BST vs Hash Table Insertion Times

- *Hypothesis*: I believe that insertions into the hash table will always be faster than the insertion into the balanced binary search tree. However, I also think that the difference in total runtime for *N* insertions into both a balanced BST and a hash table will increase exponentially as N increases.
- *Methods*: I followed these steps to get my results:
    1.) I instantiated a multiset, which represents a balanced BST, and an unordered multiset, which represents a hash table. Then, I have a loop that inserts *N* random items into each multiset and times the total insert time.
    2.) Using g++ as a compiler (with no flags), I timed the insertions for values of *N* at 100, 1000, 5000, 10000, 50000, 100000, and 500000 and placed those values in excel. For N <= 5000, I ran 100 iterations of each test and took the average of all the iterations to get more reliable data.
- *Results*:

| N | BST Insert Time (seconds) | Hash Table Insert Time (seconds) | Speed comparison HT vs BST | BST - Hash Table |
|---|---|---|---|---|
| 100 | 0.00051468 | 0.00050839 | 101.24% | 0.00000629 |
| 1000 | 0.00578984 | 0.00515516 | 112.31% | 0.00063468 |
| 5000 | 0.03137331 | 0.02586375 | 121.30% | 0.00550956 |
| 10000 | 0.0661974 | 0.0530279 | 124.84% | 0.0131695 |
| 50000 | 0.4008777 | 0.2964734 | 135.22% | 0.1044043 |
| 100000 | 0.823985 | 0.686571 | 120.01% | 0.137414 |
| 500000 | 4.791933 | 3.457707 | 138.59% | 1.334226 |






**Explanation**: The table above shows the average timing of the insertions. The last column shows as a percentage how much faster the insertions were into the hash table than the balanced BST. Below the table you can see a graph of insertion times vs *N*, with the blue curve being for the Balanced BST, and the red curve being the hash table. You also can see the equations for the linear regression lines with R^2 values. The graph below that shows how the difference in timings for insertions increases with *N*.

- *Discussion*: What I found was that both equations increase linearly with time as both linear regression equations fit with R^2 > .995. Because of this distinct and strong correlation, as *N* increases linearly, the difference in total runtime for *N* insertions into between a balanced BST and a hash table also increases linearly, whereas I hypothesized exponential increase. The equation for this is seen in figure 3 above.

- *Conclusions*: As *N* increases linearly, the runtime of inserting *N* items into a Balanced Binary search tree and a hash table also increases linearly. The equations are defined below:
    - BST: **T = 9.62e-06*x - 0.0411**
    - Hast table: **T = 6.94e-06*x - 0.0137**

  In addition, the difference in runtime between between Hash table and Balanced BST insertions increases linearly with *N*, and the equation that models this positive correlation is as follows:

- Timing Difference: **T = 2.68e-06*x + 0.0273**