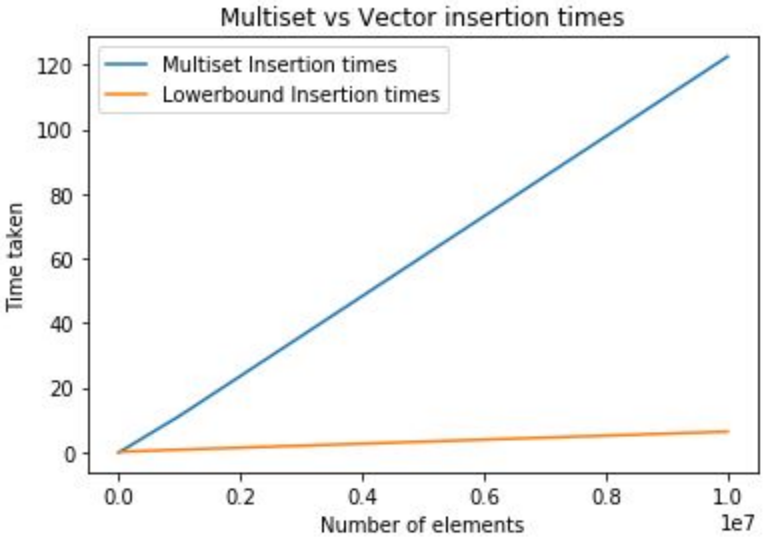
Question 4: Multiset (BST) vs Vectors

- *Hypothesis*:
  In C++, std::vector is typically considered a poor choice for a sorted container because for every insertion, all subsequent values must be shifted to make room for it. The net effect should be linear-time insertions. We predict that Multiset insertion times will be marginally slower than insertion times for a vector using binary search.
- *Methods*:
  We inserted n values into BST and found the lower bound position for sorted vector and then insert it into that position, at varying values of n, starting at 10. We then timed how long it took to run n amount of inserts, with n being multiplied by 10 every time.
- *Results*:
  We ran various experiments with different source code, and ended up concluding that all values of n, inserting into the vector was roughly 99% faster than inserting into a multiset.



n	Multiset insertion time (sec)	Vector insertion time (sec)
10	0	0
100	0.001	0
1000	0.011	0
10000	0.106	0.001
100000	1.2	0.002

1000000	10.47	0.011
10000000	129.763	0.198

- *Discussion*:
  This is not at all what we had predicted. We were very far off as we thought that inserting into a multiset was only going to be marginally slower than inserting into a vector. The time to insert into a vector never got over 2/10ths of a second, while insertion time for the multiset was exceeding 100 seconds.
- *Conclusions*:
  Under the conditions tested, insertion time for vector using binary search is always and significantly faster than inserting into a multiset.