Amin Soleimani

**Design:**
The design for both data structures are relatively the same. the difference is that now each node in the data structure has its own lock, and in order to access a node, we first need to acquire its lock. The lock used is reentrantlock, since a thread that might call a method that tries to acquire the same lock won't be blocked.
Traversing Fine-grained structures, we must acquire the lock for successor before releasing the lock for predecessor.
*Changes from CG design:*

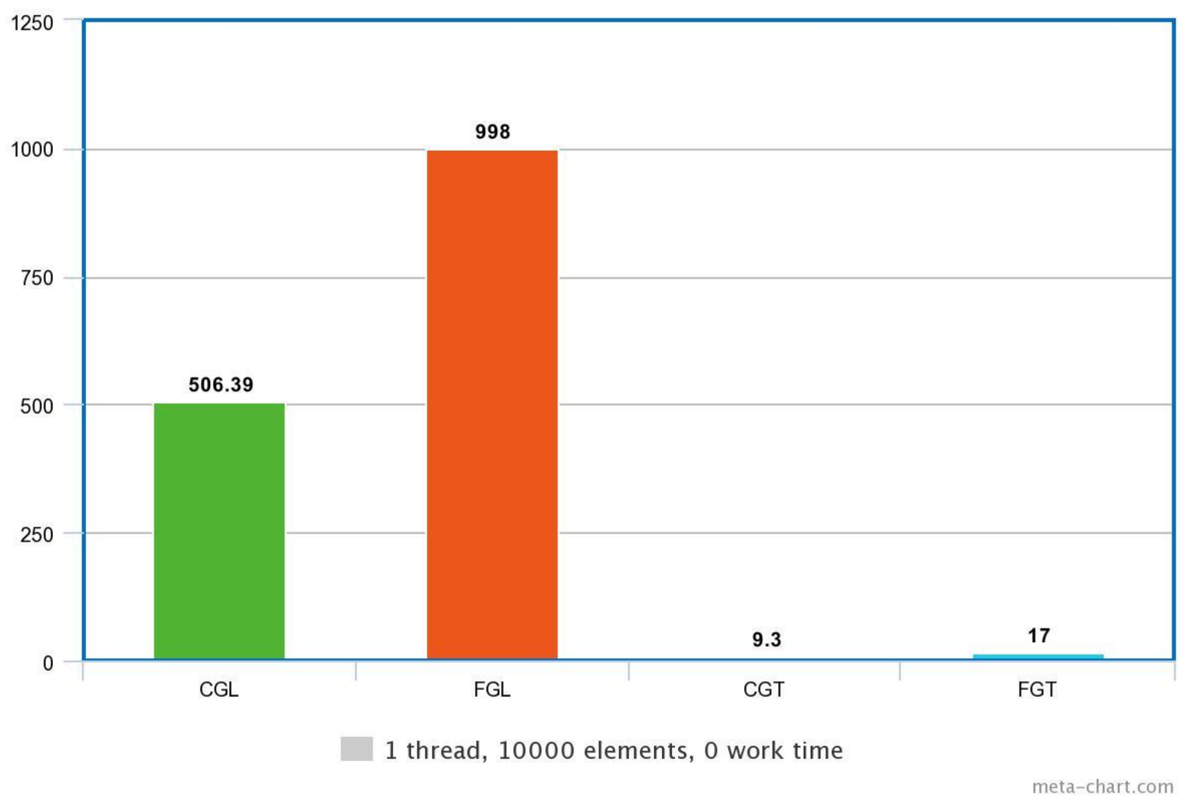
FGL:
In order to make the code more readable and avoid mistakes, I added 2 sentinel nodes. The head node initially points to the tail node, and tail's next point to NULL.
FGT:
As it was suggested in the feedback, I used a special head node. I'm using the same node class, which means the head node has 2 child nodes. The left node is always NULL, and the right node points to the root of the tree (NULL if tree is empty).
The recursive tree traversal is replaced with the iterative one.

H1:
Both coarse grained data structures should have a better performance than their fine-grained counterparts when running on 1 thread. this is due to the overhead of locking 2 nodes for accessing/modifying the data structures. This overhead should increase with the number of elements.
H2:
With the same number of threads, CGT should perform better than FGT when the total number of elements in the tree are small, since the root node and the child nodes in the first few levels would block other threads to traverse the tree. FGT performs best when tree is balanced and has the height of Log n. in the worst case, it should perform the same as FGL.

H3:
When the number of threads is high in relation to a number of elements and the time a single modification takes, increasing worktime might reduce the contention for locking the root and the top-level nodes.

Evaluation:

H1:
As it was expected, fine grained data structures have a worse performance when tasks are performed on one thread. increasing the number of items or work time increases the execution time.


CGL
FGL
CGT
FGT
1 thread, 10000 elements, 0 work time
meta-chart.com