

Homework 5 - Lists, Trees, and Tries

LinkedList vs ArrayList vs. TreeSet Timing

ArrayList

Timing

23.44 (*s*)

Memory

6.6804 (*MB*)

LinkedList

Timing

18.8428 (*s*)

Memory

12.871496 (*MB*)

TreeSet

Timing

0.007 (*s*)

Memory

15.9886 (*MB*)

Conclusion

I would say that yes, this does make sense. Since `ArrayList` has to store only the index on top of the data, it would naturally take up less storage than `LinkedList` which would have to store two pointers to the next and previous elements. It would also make sense for `ArrayList` to take more time to add all of the elements because it must double the list frequently. Finally, `TreeSet` would take only a little bit more storage since it has to store more pointers but the same amount of data. However, the adding happens almost instantly because adding for trees is $O(h)$.

Trie

Trie

Timing

0.017 (*s*)

Memory

43.033528 (*MB*)

Conclusion

I would say that this makes sense since a `trie` would have to store quite a few different nodes and the structured is mired by a lot of overhead. However, it is unbelievably fast because you only have to do work proportional to the length of the word, which means that it does not matter how many words are stored. This is different because others depend on the structure being small.