

## Project 2 Design Comparison

As implemented in this assignment, the program is required to keep the Eclipse objects ordered by catalog number at all times, which results in the use of linear search to find the location of the item to be merged. Given that there may be  $N$  elements in the file to be merged or purged, and  $N$  elements to search through in the linked list, the merge and purge operations take  $O(N^2)$  time to complete. Since duplicate elements are immediately replaced using this method, and every entry is ordered by catalog number, there is no more work to be done here.

An alternative approach of not sorting the list until a merge or purge is complete would require that the items be first appended to a linked list, an  $O(1)$  operation, and then sorted in an array, which is a  $O(N\log N)$  operation. However, this does not account for removing possible duplicate entries when merging. To do this, you need to keep track of the  $M$  items to be merged, and using the modified binary search, search by catalog number and see if you find duplicate values. This is an  $O(M\log N)$  operation, as binary search is  $O(\log N)$ , where  $N$  is the size of the linked list, and we would need to iterate through each merged item linearly, which is  $O(M)$ , where  $M$  is the number of elements to merge.

The function equation for the alternative algorithm ends up being  $O(N\log N + M\log N)$ , while the current algorithm is  $O(N^2)$ . Therefore, the alternative method is an  $O(N\log N)$  algorithm, while the implemented algorithm is  $O(N^2)$ . Even with the hidden functions it contains, the alternative method is significantly higher performing as the data set gets larger.

---

A few updates on the project:

- There are now unit tests for the LinkedList class. I wanted to get googletest working on my machine, but I couldn't get around to it on this project. Maybe next time.
- On Project 1 it was unclear as to whether the month column (column 4) was supposed to be sorted by month **number** or **lexicographically**, so I went with the latter. I am sticking with that ordering unless told to do something else.