Justin Kleiber

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(NlogN) 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(MlogN) operation, as binary search is O(logN) and we would need to iterate through each merged item linearly, which is O(M).

The function equation for the alternative algorithm ends up being O(NlogN + MlogN), while the current algorithm is a simple $O(N^2)$. Therefore, the alternative method is an O(NlogN) algorithm, while the implemented algorithm is $O(N^2)$. The alternative method appears faster in this way, but there are more steps and logical complexity in this alternative path than appear in the Big O notation.