Graham Hoehing

Nicholas Farnan

CS 1501

January 2nd, 2018

Assignment 1: Radix Sort

Why have a slow and clunky sorting algorithm for your spreadsheet application when you can use a stable and speedy sorting algorithm like LSD Radix sort. Word or numerical values, it doesn’t matter to LSD Radix sort, it will sort them all with speed and efficacy. Still unconvinced of the might of LSD Radix sort? Well, be unconvinced no longer! Let us take some other common sorting methods and see how they stack up to LSD Radix sort when it comes to our spreadsheet application. Merge sort is another common searching method that does certainly have its uses, being able to perform more complex orderings being chief among them. However, does this make Merge sort more optimal for our application than LSD Radix sort? Not by a long shot, this is because Merge sort is a comparison sort and thus like all comparison sorts, is not nearly as fast a LSD Radix sort. LSD Radix sort achieves its blazing speeds by grouping what it sorts into like groups and then sorting those groups individually rather than sorting the full set of data all at once. Now you might have heard of a sort called Quicksort, that must be very fast right? Wrong, unfortunately Quicksort, just like Merge sort, is a comparison sort and thus is naturally slower than a non-comparison sort like LSD Radix sort. The same can be said for Shell sort, Bubble sort and any of the other comparison sorts. That’s not to say these sorts do not have their uses, but for our spreadsheet application there can only be one real choice, LSD Radix sort.

LSD Radix sort or Least Significant Digit Radix sort is so fast and stable because when sorting keys of the same length are grouped together and then the LSD Radix sort is performed on each of these group separately, the allows LSD Radix support to avoid going over the full list of keys with each sorting pass. As the name would suggest, LSD Radix sort starts sorting from the least significant digit of the data set and works from there. This allows LSD Radix support to achieve a worst case Big-O run time of O(nw), where n is the number of keys and w is the average length of those keys, nothing to scoff at. This grouping of sorting keys is also what makes LSD Radix sort such a stable and reliable sort. A stable sort algorithm orders identical values the same way they were ordered in the input, for example if your data had two ‘6’s in it one appearing in the very beginning of the data, and the other much farther down the line, after the sort the two would be next to each other, but the order would be maintained. Merge sort is far slower than LSD Radix sort with a Big-O runtime of n log n. The worst-case Big-O runtime of any comparison sort like Merge sort will never be faster than n log n, and some have even slower worst-case runtimes, such as Quicksort with a worst case Big-O runtime of (n\*n). These slow runtimes mean our spreadsheet application will be a lot slower, as each time the sort is run the difference will be more noticeable, using a comparative sort would be much less practical than if it was being run using a LSD Radix. The LSD Radix sort allows us to sort our data much faster than any of the comparison-based sorts, such as shell sort or bubble sort.