Daesang Yoon

CS 1501 – Essay 1

Dr. Nicholas Farnan

9/15/2017

First Part:

There are lots of sorting algorithms which we can select to use in future spreadsheet application. There are bubble sort, selection sort, insertion sort, shell sort, merge sort, quick sort, radix sort, bogo sort, etc. In fact, we can choose any one sorting algorithm from the selection above for our spreadsheet application. Each sorting algorithm has its own disadvantages and advantages. For example, bubble sort is very easy to implement and the process of how it works is easily comprehensible to the audience who do not have a deep understanding or some experiences in programming. However, bubble sort is not generally a good idea because it will take tremendous running time to sort completely. It may take infinitely depending on the database or elements need to be sorted. Also, President Obama acknowledged that bubble sort would be the wrong method to sort a million 32-bit integers or huge amount of data.

We can still choose another different approach and sorting algorithm. In fact, there is a more practical and general sorting algorithm such as merge sort. Merge sort is far the better choice than bubble sort because it usually guarantees to have good worst case performance. Its time complexity is usually n log n for an array of n items. However, merge sort does not sort in place. It requires an extra copy array which will take extra space or memory.

Now I want to introduce a different sorting algorithm which is radix sort. Radix sort is obviously more efficient than simple sorting algorithms like bubble sort. Also, radix sort has a very important property which is stability. Specifically, least-significant-digit (LSD) radix sorts your spreadsheet application stably and can have faster performance in certain circumstances than comparison-based sorting algorithms, such as quick sort, merge sort, etc.

Second Part:

LSD (least-significant-digit) radix sort is the method originally used for sorting punched cards by cards-sorting machines that were invented in the beginning of the 20th century. These old punched cards sorting machines were capable of distributing a deck of punched cards among 10 bins by the pattern of holes punched in the selected columns. If the cards have numbers punched in a particular set of columns, a person sorts the cards by running them through the machine on the rightmost digit, then stacking the output decks in order and running through the machine on the next rightmost digit, and so forth, until proceeding to the leftmost digit. The stacking part of the cards is a stable process which uses key indexed counting sort. This version of LSD radix sorting took not only a significant role in commercial applications in the past but it was also popular among many cautious programmers. LSD radix is significant because it is a linear time sort for typical applications.

Most general sorting algorithms are comparison-based sorts which denote an algorithm that has only one way of obtaining information, which is by comparing two elements to see the smaller one and higher one. However, radix sort is different. Radix sort, a non comparison-based sort, does not compare two elements to sort the data, but it sorts the data by grouping keys by the digits which share the same position and value. These keys can be a string of characters or numerical digits in a given radix or base.

In LSD radix sorting, short keys come before longer keys, and keys of the same length are sorted lexicographically. Typically, the process is that each key is first dropped into a bucket corresponding to the value of the least-significant-digit. A bucket keeps the original order of the keys as they are dropped. This process of preserving the original order is stable sorting which makes LSD radix sort a good fit for the spreadsheet application. Even though the data has to be sorted, one may want to keep the original order of the data for the safety measure, and LSD radix sort allows conserving the original order.

Then, we jump to next more significant digit and group the keys according to the value of the current digit. We continue to repeat the process until we do not have more digits to proceed. Also, the keys are sorted lexicographically so that users can look at neatly dictionary-ordered data when using spreadsheet application. This feature can also be beneficial to many spreadsheet applications and users.

A general and efficient comparison-based sorting algorithm such as merge sort has worst-case performance of O (n log n) when sorting n objects. The performance is much better than simple sorting algorithms and merge sort guarantees to maintain same performance. However, comparison-based sorts (merge sort) cannot perform better than O (n log n) running time. Because comparison sorts cannot do better than its limit, LSD radix sort is considered as a good alternative to some comparison-based sorts. LSD radix sort complexity time is O (*w*n), where n is number of keys and *w* is average length or word size (a number of bits in word). By using LSD radix sort, we can sort the data in linear time. Given that word size is less than log n, LSD radix sort will perform faster than comparison sorts. Radix sort has limitations on the data type of keys that need to be sorted because it needs radix so the sort is typically used for string and integer keys. However, the spreadsheet data are usually stored as alphabet characters and numbers and LSD radix sort can perform better than comparison-based sorting algorithm as long as word size or key range is not large.