Sebastian Rock

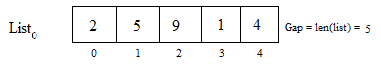
Professor Kopp

Data Structures

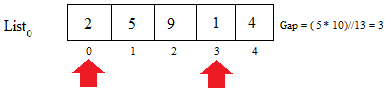
6 November 2013

Comb Sort Evaluation

The comb sort works by comparing two locations in a list by a gap.



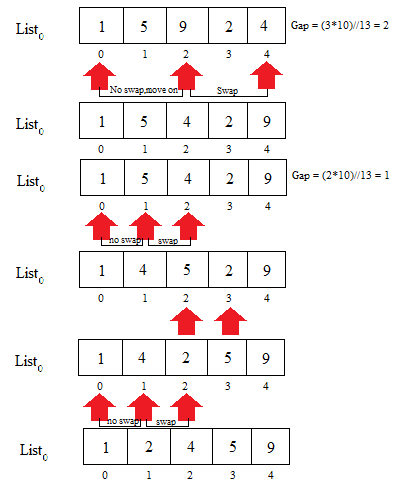
The comb sort starts by obtaining the length of the list, and then comparing the two values. For any gap n, the algorithm compares the value stored in the first location with a list to value stored the index n. In this case, the gap starts at 5. The comparisons, however, don’t start until the gap is updated at least once.



If the number stored there is less than the number stored in the first location, it switches locations. The comparison then either compares the value stored at index n to the value stored at index n + n, or it updates the gap if the list doesn’t have n + n elements. Since the gap is 3, the comparison starts at position 0 and goes to position 3 and compares them. Since 2 is greater than 1, they switch positions.



This continues until all positions in the list have been swapped by the current gap and the gap is less than 1.



In structure, the comb sort is similar to the bubble sort. The big difference is that the comb sort calculates a shrink factor ((gap \* 10) // 13), while the bubble sort has a constant rate of change for the shrink factor. Because of this, the comb sort is able to more efficiently sort through lists. Another advantage of the comb sort is that it can more quickly deal with small values at the end of a list, which is something that tends to slow down the bubble sort. As a disadvantage, however, it tends to be much slower when the list size is fairly big. Although it is superior to the bubble sort, it’s still a slower method of sorting and should be used sparingly.

Results

The big-Oh notation for the comb sort is similar to that of the bubble sort. The worst case scenario is O() and the best case scenario is O(n). In practice, however, the comb sort performed the worst out of all the methods of sorting. The bubble sort performed the fastest, with the insertion coming in second, the selection coming in third, and the comb sort finishing last. Theoretically, the comb sort should have outperformed the bubble sort in larger lists, but it severely underperformed in all tests.