Merge Sort

CS32

By Freddy Yang

Basic Idea

Data Structure=Arrays/Linked List

Steps (with arrays):

1. Divide the data into different groups of arrays with possibly the smallest number of elements in each array (which is 2 in most cases)
2. Sort each array

3. Merge every pair of adjacent arrays into a new one by comparing the first element of the first array with each element of the second array and then the second element of the first array with each element that's left in the second array.
4. Repeat Number 3 until they are all merged into one single array with sorted data in it.

Running Time

- Best Case: O (n log n)
- Worst Case: O (n log n)
- Average Case: O (n log n)

Example 1

3,5,4,2

-> 3,5   4,2 (divide into two groups)

-> 3,5   2,4 (sort each one)

-> 2,3,4,5 (from 3,5 and 2,4, compare 3 and 2, put the smaller one in the new array; then compare 3 and 4, and do the same thing)

Example 2

2,6,3,1,4,8,7,5

->2,6   3,1   4,8   7,5 (divide into arrays)

->2,6   1,3   4,8   5,7 (sort each one)

->1,2,3,6   4,5,7,8 (merge each two pairs)

->   1,2,3,4,5,6,7,8

Example 3

- German Folk Dance Video

http://www.youtube.com/watch?v=XaqR3G_NVoo

Notes

-Most merge sort does not run in-place.

-In order to run in-place, or without extra memory, recursions have to be used in a merge sort.

Sources

- http://www.personal.kent.edu/~rmuhamma/Algorithms/MyAlgorithms/Sorting/mergeSort.htm

- http://www.algolist.net/Algorithms/Merge/Sorted_arrays

- http://stackoverflow.com/questions/2571049/how-to-sort-in-place-using-the-merge-sort-algorithm

- http://stackoverflow.com/questions/7801861/why-is-merge-sort-worst-case-run-time-o-n-log-n