Joseph Garcia

3/21/2020

CS 413

Homework 3 Report

Counting Inversions Algorithm

The counting inversions algorithm counts the number of inversions (array changes) that are made before the array is fully sorted. No inversions are made if the array is already sorted. The merge sort algorithm is used for sorting the array. Dividing the array uses the recurrence relation of 2T(n/2), while merging takes O(n) time. The merge sort algorithm uses the recurrence relation of T(n) = 2T(n/2) + O(n). Using Master Theorem case 3, the time complexity of merge sort is O(nlog(n)).

Merge Sort Algorithm

The merge sort algorithm recursively divides the array into two halves. Each of those halves are then divided again, resulting in more and more divided parts. Once all the parts have been divided into individual units, the units start to merge back together, with each merged part having the values sorted in descending order. Eventually, all the parts are merged back into one array, with all the values sorted in descending order. Dividing the array uses the recurrence relation of 2T(n/2), while merging takes O(n) time. The merge sort algorithm uses the recurrence relation of T(n) = 2T(n/2) + O(n). Using Master Theorem case 3, the time complexity of merge sort is O(nlog(n)).