Data Structures 11/18/2016

0145-343-001

Note Taker: Jai Punjwani

ANNOUNCEMENTS

* Next HW: <http://home.adelphi.edu/~siegfried/cs343/343hw9.html> (heap sorting - due Monday)

Notes:

PowerPoint: <http://home.adelphi.edu/~siegfried/cs343/343l6.pdf>

Topic: Sorting (cont’d)

**Merge Sort:** divide-and-conquer approach. Algorithm divides list of size n into n sublists, and repeatedly merges them, and sorts on each merge. So initially, the list is divided into n sublists, and then they become n/2 sublists, as the sort takes place on each step, and so on. The complexity of merge sort is ***O***(n log n).

**Insertion Sort:**  sorting method in which the list is broken into a list is broken down into sub-lists of the first 1, 2, … and finally n elements. Each sub-list is sorted, and when the next element is added, it is only a matter of placing that element in the position it belongs in. The complexity is ***O***(n2). This is faster than the bubble sort because it makes fewer interchanges.

**Shell Sort:** A type of insertion sort in which items that are far apart to be sorted, whereas a standard insertion sort only allows sorting of elements in order. The complexity of this is more than ***O***(n log n) but less than ***O***(n2). Namely it is ***O***(n (log n)2).

**Other Sorts –** radix sort (see ppt), cocktail sort\*

\*(<https://www.youtube.com/watch?v=IAADdYrts2o>)