Data Structures 11/11/2016

0145-343-001

Note Taker: Jai Punjwani

ANNOUNCEMENTS

* QUIZ – Wednesday 11/16/2016 (binary and general tree traversal)
* Next HW: <http://home.adelphi.edu/~siegfried/cs343/343hw8.html> (inventory using a binary tree) – due November 24th

Notes:

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

Topic: Sorting

**Quick sort** – divide and conquer sort. Basically, the algorithm divides an array into two smaller sub-arrays by first choosing a pivot, and then moving all elements less than the pivot into the array on the left, and all elements greater than the pivot into the array on the right. Elements that are equal to the pivot, and the pivot itself can go in either array. Then, for each array we choose another pivot and repeat the process recursively.

Complexity: ***O***(n log n) [on average]

***O***(n2) – worst case

**Selection sort** – inefficient in-place comparison sort. [***O***(n2) is best case]; this may be better than a bubble sort in some cases because it does not move around elements that may already be in the right spot on each iteration.

**Heap sort** – heap is a tree-based data structure. In this algorithm, we try to obtain a sorted heap in which the father is greater than its sons (each root is greater than its sons). First we organize the array into a heap, and then we repeatedly take the first element and place it on the correct position on the heap. It has complexity ***O***(n log n) in its WORST case, and ***O***(n) in the best case.