Design and Analysis of Algorithms
CS575, Spring 2023

Theory Assignment 2.2

Due on 3/7/23 (Tuesday)

1. (15 points) We want to find the largest item in a list of n items.
   a) Use the divide-and-conquer approach to write an algorithm (pseudo code is OK). Your algorithm will return the largest item (you do not need to return the index for it). The function that you need to design is *int maximum (int low, int high)*, which *low* and *high* stands for low and high index in the array. (8 points)
   b) Analyze your algorithm and show its time complexity in order notation (using θ). (7 points)

2. (15 points) Illustrate the operation of Heapsort on the input array A = <5, 13, 2, 25, 7, 17, 20, 8, 4>. Draw the heap just after Build-Max-Heap was executed. Then draw a new heap after another (the next) element has been sorted; the last heap you draw has a single element (see example figure in slide 24 of Ch6-sorting-heap-linear lecture notes). Ex 6.4-1 Pg 172

3. (15 points) Argue for the correctness of Heapsort (the slide 25 of Ch6-sorting-heap-linear lecture notes) using the following loop invariant: At the start of the iteration with an *i* of the for loop, (a) the subarray $A[1 .. i]$ is a max-heap containing the *i* smallest elements of $A[1 .. n]$, and (b) the subarray $A[i+1 .. n]$ contains the $n - i$ largest elements of $A[1 .. n]$ in correctly sorted order (i.e., in ascending order). Divide your proof into the three required parts: Initialization, Maintenance, and Termination.