CSci 242: Algorithms and Data Structures **Spring, 2020**

Instructor: Dr. M. E. Kim Date: February 21st, 2019

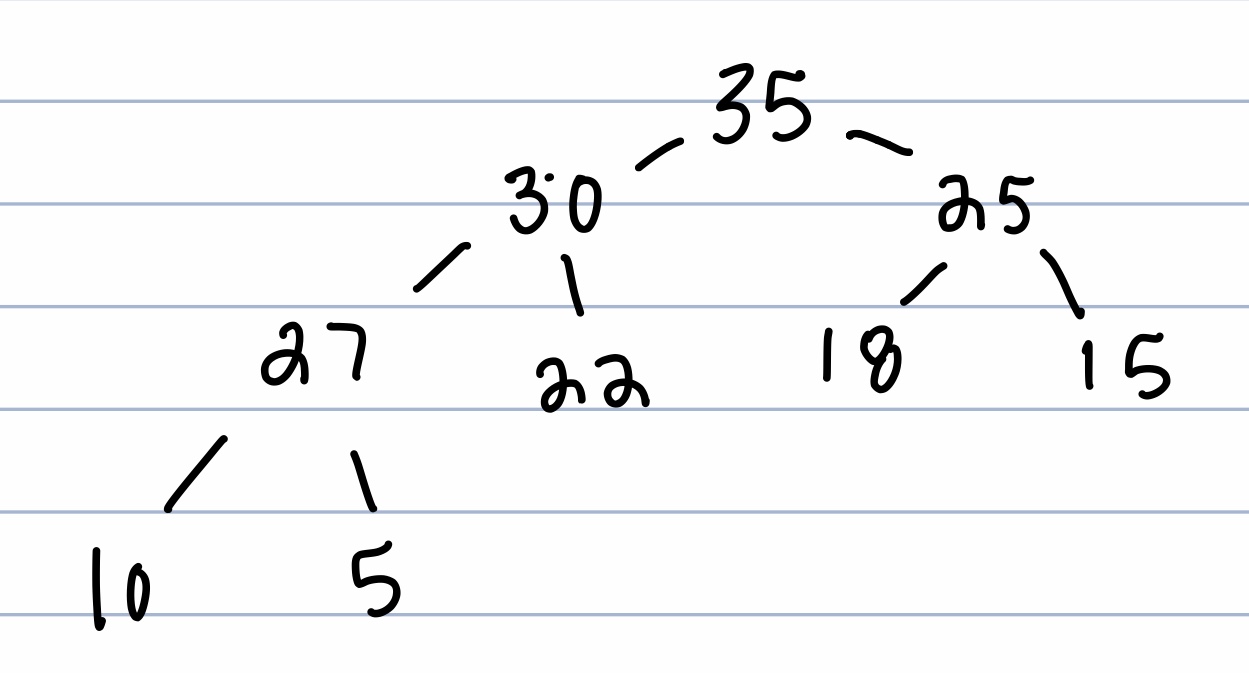
Due: by the end of day, February 29th (Sat.), 2020.

**Home Assignment 3B: 30/40 points**

Q5. [30] **Maximum Heap**

In the array A[1..10] that stores the elements [35, 30, 25, 27, 22, 20, 18, 15, 10, 5],

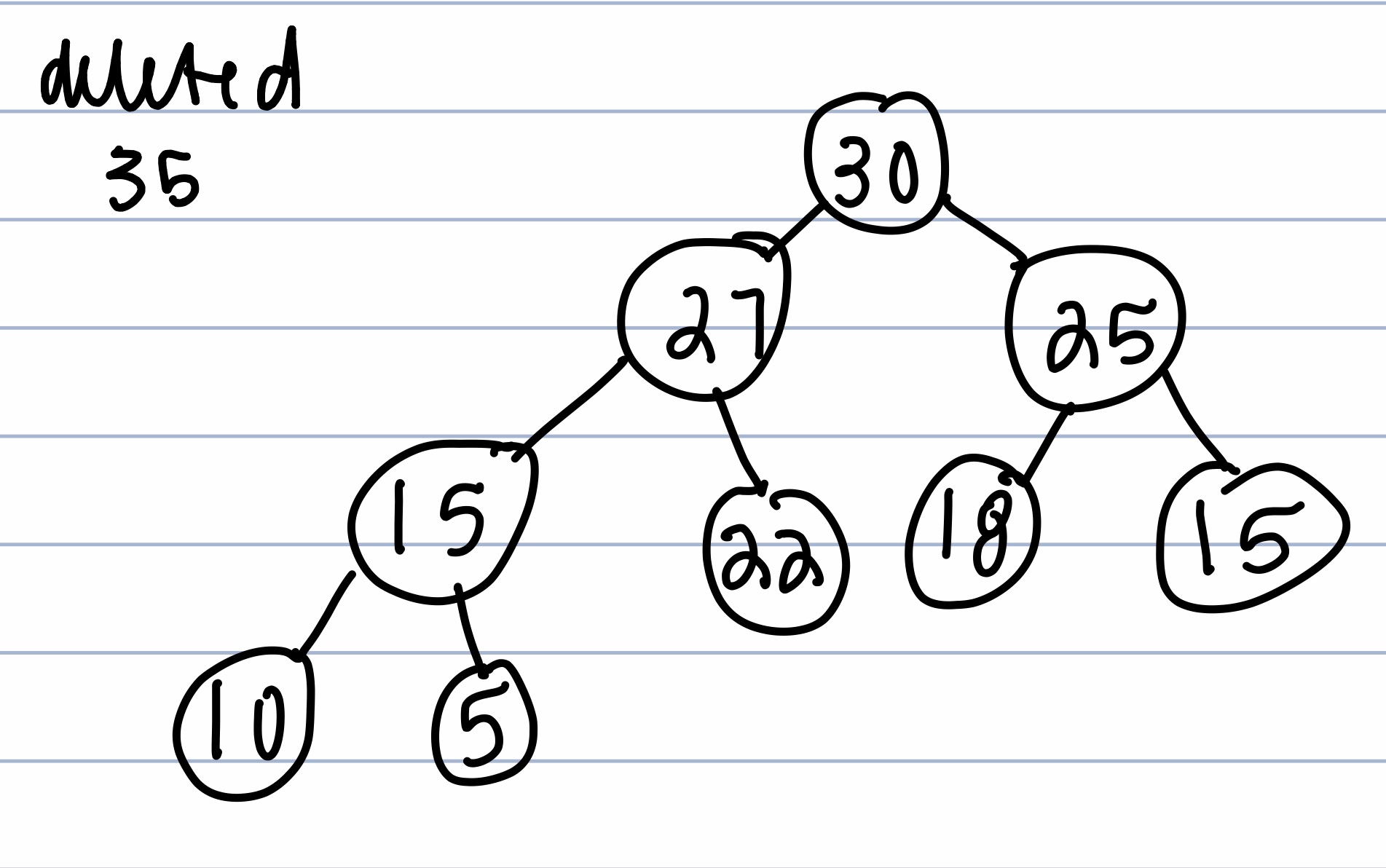
1. [9/10] ***Construct*** a ***max-heap*** in the array A. Draw the final heap and show its content in the array A. During the construction of heap, show each step of heap construction.



The node 20 is missing from the above.

[35, 30, 25, 27, 22, 20, 18, 15, 10, 5] itself is a max-heap.

1. [9/10] Draw the max-heap after ***removing the Maximum key*** from 1) and show its content in the array A.



Solution:

[~~35~~, 30, 25, 27, 22, 20, 18, 15, 10, 5] 🡪 swap

[5, 30, 25, 27, 22, 20, 18, 15, 10, | ~~35~~] 🡪 restore the max-heap property

[30, 27, 25, 15, 22, 20, 18, 5, 10] -- final heap

1. [7/10]From the max-heap in 2), ***sort t***he array A in the ***ascending*** order. Draw the heap and show the content of array A***after sorting five largest keys.***

AlgoirthmSortAscending

For I = size of (A) to 1 do

A(i) <- removeMax()

i++

Return A

Result : [5, 10, 15, 18, 20, 22, 25, 27, 30, 35]

First 5 elements : [5, 10, 15, 18, 20]

Solution:

[ 20, 18, 5, 15, 10 | 22, 25, 27, 30, 35]

Or [18, 15, 5, 10 | 20, 22, 25, 27, 30 | (35) ]

Q6. [5/10] **Heap Sort**

A minimum heap is used to sort an array A of length *n* in descending order.

What is the running time of Heap-Sort algorithm on an array A that is already sorted in descending order? Explain your answer.

The run time for the algorithm will be O(n) because the contractions of the heap will take O(1), removing the max will take O(1). Thus as a result by running both as O(1) on n size will take O(n).

Solution:

O(*n* log *n*) – Regardless the state of sorting, you have to (re)construct a minimum-heap to do heap-sort, breaking the original sorted order. Thus, the sorted order of input doesn’t affect the running time of heap-sort.

i.e. the running time is still O(*n* log *n*).