**Sam Lazrak**

**CS 303 Algorithms and Data Structures**

**Homework Assignment 4**

**2/6/18**

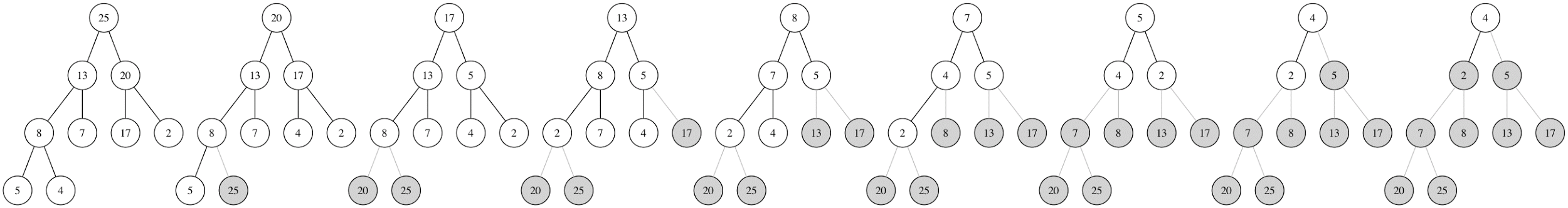
**Work the following Exercises from Chapter 6 of the text:**

* **A. 6.1-1: What are the minimum and maximum numbers of elements in a heap of height h?** 
  + “Since a heap is an almost-complete binary tree (complete at all levels except possibly the lowest), it has at most 1+2+22 +23 +…+2h =2h+1 -1 elements (if it is complete) and at least 2h-1 +1=2h elements (if the lowest level has just 1 element and the other levels are complete)” ("Homework 3 Solutions", 2013).
* **B. 6.1-2: Show that an n-element heap has height .**
* “The number of internal nodes a complete binary tree has is 2h−1 where h is the height of the tree. A heap of height h has at least one additional node (otherwise it would be a heap of length h−1) and at most 2h additional nodes (otherwise it would be a heap of length h+1). Thus, if n∈(2h,2h+1−1), then the height will be ⌊lgn⌋” (Kanev).
* **C. 6.4-1: Using Figure 6.4 as a model, illustrate the operation of HEAPSORT on the array A{5,13,2,25,7,17,20,8,4}. Models:** (Kanev)

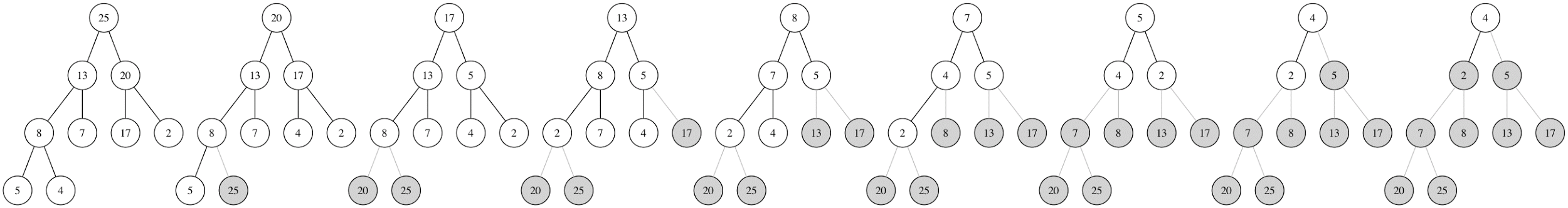
(a) is the max-heap data structure just after BUILD-MAXHEAP has built it in line 1 of the heapsort algorithm.

(b) – (i) is the max-heap just after each call of MAX-HEAPIFY in line 5 of the heapsort algorithm.

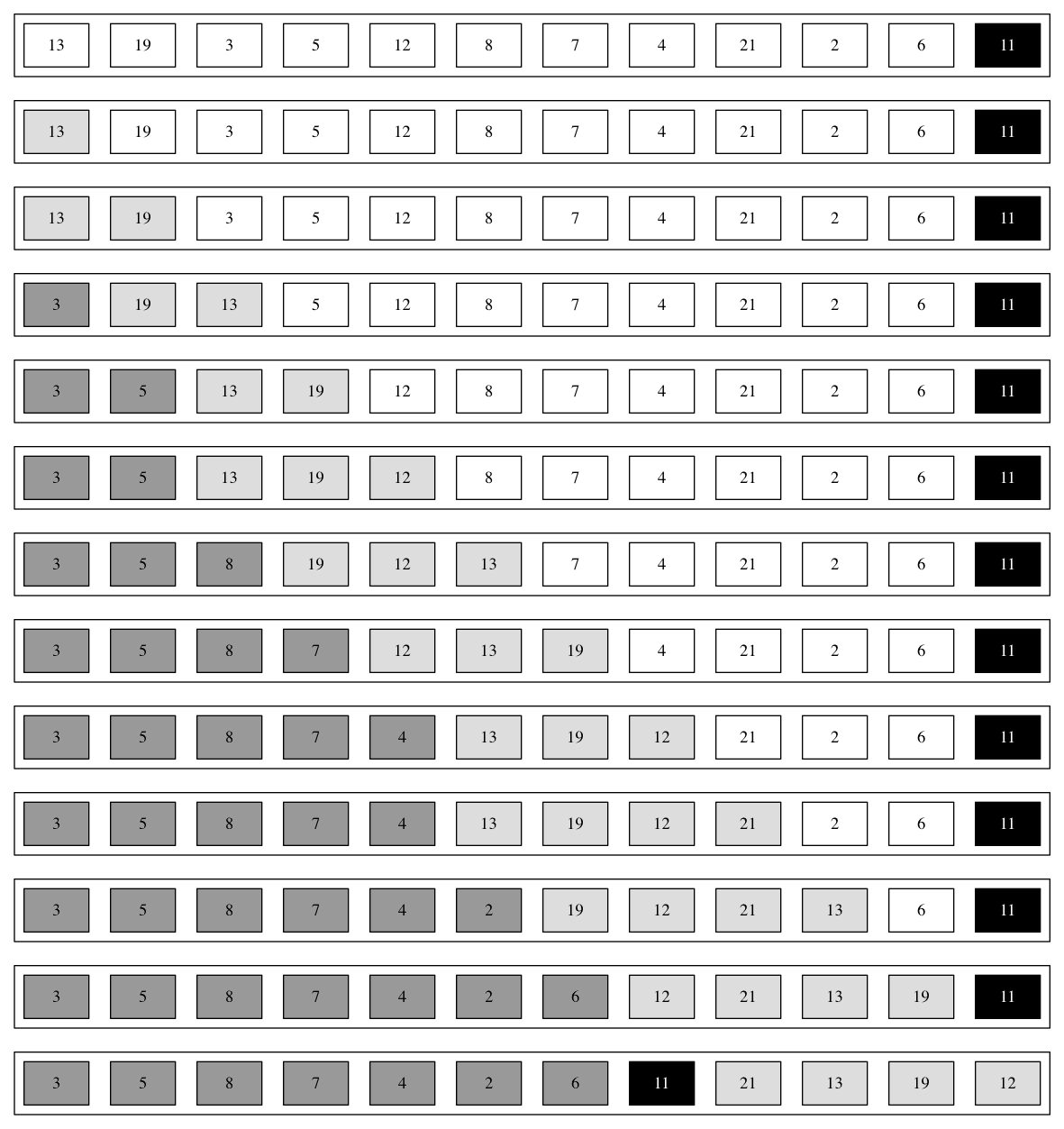
(a) (b) (c) (d) 



(f) (g) (h) (i)



* **D. 6.4-3: What is the running time of HEAPSORT on an array A of length n that is already sorted in increasing order? What about decreasing order?**
* Both of them are Θ() (Kanev).

1. **7.1-1: Using Figure 7.1 as a model, illustrate the operation of PARTITION on the array A{13,19,9,5,12,8,7,4,21,2,6,11}.**

(Kanev)

1. **7.2-2: What is the running time of QUICKSORT when all elements of array A have the same value?**

* When all elements are equal, the operation of partition will evaluate each putting it in the left partition each subsequent time. One of the partitions is always empty leading to unbalanced partitioning. The running time is equivalent to the worst case running of quicksort because it occurs when the partitioning routine produces one sub problem with n-1 elements and one with 0 elements. Each recursive call will lead to unbalanced partitioning. The partitioning costs Θ(n) time. Thus the recurrence will be: T(n) = T(n−1) + Θ(n). Evaluating that with the arithmetic series, the running time of quicksort in this case is Θ(𝑛2) (Cormen).

1. **7.2-4: Banks often record transactions on an account in order of the times of the transactions, but many people like to receive their bank statements with checks listed in order by check number. People usually write checks in order by check number, and merchants usually cash them with reasonable dispatch. The problem of converting time-of-transaction ordering to check-number ordering is therefore the problem of sorting almost-sorted input. Argue that the procedure INSERTION-SORT would tend to beat the procedure QUICKSORT on this problem.**

* “INSERTION-SORT’s running time on perfectly-sorted input runs in Θ(n) time. So, it takes almost Θ(n) running time to sort an almost-sorted input with INSERTION-SORT. However, QUICKSORT requires almost Θ(𝑛2) running time, recalling that it takes Θ(𝑛2) time to sort perfectly-sorted input. This is because when we pick the last element as the pivot, it is usually the biggest one, and it will produce one sub problem with close to n – 1 elements and one with 0 elements. Since the cost of PARTITION procedure of QUICKSORT is Θ(n), the recurrence running time of QUICKSORT is T(n) = T(n – 1) +Θ(n). In another problem, we, use the substitution method to prove that the recurrence T(n) = T(n – 1) +Θ(n) has the solution T(n) = Θ(𝑛2). So we use INSERTION-SORT rather than QUICKSORT in this situation when the input is almost sorted” ("Homework 3 Solutions", 2013).

1. **7.3-1: Why do we analyze the expected running time of a randomized algorithm and not its worst-case running time?**

* “The worst-case running time is not triggered by a specific output, but occurs randomly. We're not interested in it, since we cannot reproduce it reliably. Instead, it is factored in the analysis of the expected running time” (Kanev).

Works Cited

Cormen, Thomas H. *Introduction to Algorithms*. Cambridge, Mass: MIT Press, 2001. Print.

"Homework 3 Solutions." *Texts in Computer Science Theory of Computation* (n.d.): 326-28. 2013. Web.

Kanev, Stefan. *Introduction to Algorithms*. N.p., n.d. Web.