Algorithms* Lab 3

1 Review

Topics covered this week:

- recurrences
- heaps and heapsort

2 In lab exercises COLLABORATION LEVEL: 0

1. (CLRS 6.1-1) What are the minimum and maximum number of elements in a heap of height h? Note: the height of a heap is the number of edges on the longest root-to-leaf path.
2. (CLRS 6.1-2) Show that an n-element heap has height $\Theta(\lg n)$ (more precisely, $\lceil \lg n \rceil$).
3. (CLRS 6.1-3) Where in a min-heap might the largest element reside, assuming that all elements are distinct?
4. (CLRS 6.1-5) Is an array that is in sorted order a min-heap?
5. (CLRS 6.1-7) Argue that the leaves are the nodes indexed by $\lfloor n/2 \rfloor + 1, ..., \lfloor n/2 \rfloor + 2, ..., n$.
6. What is the effect of calling HEAPIFY(A, i) for $i > size[A]/2?$. Here i is the index of the node where HEAPIFY is called.
7. (CLRS 6.5-2) Illustrate the operation of HEAP-INSERT (A, 7) on the heap (note: this is a min-heap):
$$A = \{2, 5, 10, 6, 8, 100, 11, 9, 15, 9, 10, 200, 101\}$$
8. (CLRS 6.2-1) Illustrate the operation of HEAPIFY(A, 1) on
$$A = \{(20, 5, 10, 6, 8, 100, 11, 9, 15, 9, 10, 200, 101, 12\}$$
9. (CLRS 6.3-1) Illustrate the operation of BUILD-MAX-HEAP on the array
$$A = \{5, 3, 17, 10, 84, 19, 6, 22, 9\}$$
10. (CLRS 6.4-1) Illustrate the operation of Heapsort on the array
$$A = \{5, 13, 2, 25, 7, 17, 20, 8, 4\}$$
11. (CLRS 6.4-3) What is the running time of Heapsort on an array of length n that is already sorted in increasing order? What about decreasing order?

*csci2200, Laura Toma, Bowdoin College

3 Homework COLLABORATION LEVEL : 1

- **Grading:** The assignment will be evaluated based not only on the final asnwer, but also on clarity, neatness and attention to details.
- **Writing:** Please write each problem on a separate sheet of paper, and write your name on each sheet. The problems will be graded by different TAs.

1. Come up with an algorithm that finds the kth smallest element in a a set of n distinct integers in $O(n + k \lg n)$ time.
2. (C-4.9) Suppose we are given a sequence S of n elements, each of which is colored red or blue. Assuming S is represented as an array, give an $O(n)$ and in-place method for ordering S so that all blue elements are listed before all the red elements.
3. (CLRS 6.5-9) Assume you have k sorted lists containing a total of n elements, and you want to merge them together in a single (sorted) list containing all n elements. For simplicity you may assume that the k lists contain the same number of elements.
   (a) Approach 1: merge list 1 with list 2, then merge the result with list 3, then merge the result with list 4, and so on. What is the worst-case running time ?
   (b) Approach 2: split the k lists into two halves, merge each one recursively, then use the standard 2-way merge procedure (from mergesort) to combine the two halves. What is the worst-case running time ?
   (c) Give another approach (to merge the k lists) that uses a heap, and runs in $O(n \lg k)$-time.
4. (CLRS 7-3) Professors Dewey, Cheatham, and Howe have proposed the following "elegant" sorting algorithm:

   STOOGE-SORT(A, i, j)
   if $A[i] > A[j]$
       then exchange $A[i] \leftrightarrow A[j]$
   if $i + 1 \geq j$
       then return
   $k \leftarrow \lfloor (j - i + 1)/3 \rfloor$
   STOOGE-SORT$(A, i, j - k)$
   STOOGE-SORT$(A, i + k, j)$
   STOOGE-SORT$(A, i, j - k)$

   a. Argue that STOOGE-SORT$(A, 1, \mathit{length}[A])$ correctly sorts the input array $A[1..n]$, where $n = \mathit{length}[A]$.
   Hint: Argue that it sorts correctly any array of 1 or 2 elements. Then assume that it sorts sorrectly any array of $2n/3$ elements, and argue that this implies that it sorts correctly any array of n elements (What is true after the first recursive call? After the second?)

   b. Give a recurrence for the worst-case running time of STOOGE-SORT and a tight asymptotic (Θ-notation) bound on the worst-case running time.