Worksheet: Heaps

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 6.5-3) Write pseudocode for the procedure HEAP-INSERT on a min-heap.
9. (CLRS 6.2-1) Illustrate the operation of HEAPIFY(A, 1) on
$$A = \{(20, 5, 10, 6, 8, 100, 11, 9, 15, 9, 10, 200, 101, 12\}$$
10. (CLRS 6.3-1) Illustrate the operation of BUILD-MAX-HEAP on the array
$$A = \{5, 3, 17, 10, 84, 19, 6, 22, 9\}$$
11. (CLRS 6.4-1) Illustrate the operation of Heapsort on the array
$$A = \{5, 13, 2, 25, 7, 17, 20, 8, 4\}$$
12. (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?
13. How would you implement a function that searches for a given element in a heap, and how long would it take in the worst case?
14. (GT C-2.31) Develop an algorithm that computes the kth smallest element in a set of n distinct integers in $O(n + k \lg n)$ time.