cs201c: Tutorial 4
Binary, Binomial, and Leftist Heaps

Note. All heaps are min-heaps.

1. Let H be a binary heap. Suppose you are given a key k. Find all keys <= k in heap H in O(m+1) time, where m is the number of such keys.

2. Let k be an integer greater than or equal to 2. A k-ary heap is a complete k-ary tree. (Each non-leaf node in a k-ary tree has at most k children.) A binary heap is exactly a 2-ary heap. Describe insert, deleteMin, and decreaseKey (given a pointer to the node) in a k-ary heap and bound their worst-case running times as exactly as possible.

3. Show that a binomial heap on n given keys can be built in O(n) time. You can only use a sequence of merge operations, starting from an empty binomial heap.

4. Show that a leftist heap on n given keys can be built in O(n) time. You can only use a sequence of merge operations, starting from an empty leftist heap.

5. Let H be a leftist heap on n keys with a right spine of length h. What is the maximum possible length of a root-to-leaf path in H.