IT623 Lab 11 - Max Heaps

Please refer to the Java program from Lafore posted here (heap.java). The `Heap` class implements a binary max heap. It uses a `Node` class whose only field is a variable that serves as the node's key. The `Heap` class contains the methods we discussed in class, plus `isEmpty()` and `displayHeap()`, which outputs a crude but comprehensible character-based representation of the heap. Lafore puts the root at index 0 to save one element, which changes things slightly. If `i` is the index of a node, then the indices of its parent, left child, and right child can be computed as `(i-1)/2`, `2*i + 1`, and `2*i + 2`, respectively. The `main()` routine in `HeapApp` creates a heap with a maximum size of 31 (dictated by the limitation of the display routine) and inserts into it 10 nodes with random keys. Then it enters a loop in which the user can enter `s`, `i`, `r`, or `c`, for show, insert, remove, or change.

1. Extend the Heap class to include a method `findMin()` that returns the element with the smallest key from a max-heap. Your method should use only $\lceil n/2 \rceil$ element comparisons for an n-element heap. Modify the Heap class so that the user can enter t, to execute this operation.
2. Extend the Heap class to include a method `sort()` that sorts the elements in the heap in ascending order, that is, from smallest to largest. Modify the Heap class so that the user can enter o, to execute this operation.

No submission is required for this lab. However you must get your work checked by a lab TA.
