1. In a previous recitation, we discussed the priority queue abstract data type. Max priority queues support two operations:

   - INSERT(x) - Add the item x to the queue. The value of x must be smaller than all previously extracted values.
   - EXTRACT-MAX() - Delete (and return) the x with the highest value from the queue.

   Describe how to implement a priority queue using a max-heap. Assume that the maximum number of elements in the queue at any point is k, and this value is known beforehand. What is the worst-case running time of the INSERT and EXTRACT-MAX operations (in terms of the number of items in the priority queue)?

2. The MERGE operation of MERGE-SORT merges two lists of length n in $\Theta(n)$ time. Let's implement K-MERGE, a generalized version of the MERGE operation that merges k lists instead of 2 lists.

   (a) If we implemented K-MERGE using the same process as the old MERGE (that is, repeatedly finding the minimum element at the front of each list and removing it), how long would K-MERGE take?

   (b) Can we implement K-MERGE more efficiently? HINT: use heaps.