1. Show that, for any \( n \), there is a sequence of insertions in a max-heap that requires \( \Omega(n \log n) \) time to process.

2. Consider a max-heap \( T \) for storing \( n \) keys. Give an efficient algorithm for reporting all the keys in \( T \) that are greater than or equal to a given query key \( x \) (which is not necessarily in \( T \)). Your algorithm should run in \( O(k) \) time, where \( k \) is the number of keys reported.

3. A \textit{Staq} is a data structure combining properties of both stacks and queues. It can be viewed as a list of elements written left to right such that three operations are possible:
   - \texttt{STAQPush}(x): add a new item \( x \) to the left end of the list.
   - \texttt{STAQPop}(): remove and return the item on the left end of the list.
   - \texttt{STAQPULL}(): remove and return the item on the right end of the list.

   Implement a Staq using \textit{three} stacks and \( O(1) \) additional memory so that the amortized time for any \texttt{STAQPush}, \texttt{STAQPop}, or \texttt{STAQPULL} operation is \( O(1) \). In particular, each element in the Staq must be stored in exactly one of the three stacks. Again, you are only allowed to access the component stacks through the interface functions \texttt{PUSH} and \texttt{POP}.

4. In class we studied \textit{binary} Huffman coding of the letters in the alphabet \( S \). Generalize Huffman’s algorithm to create a ternary encoding of the letters in \( S \).
5. (a) Using Huffman encoding scheme on a set $S$ of $n$ symbols with frequencies $f_1, f_2, \ldots, f_n$, what is the longest a codeword could possibly be? Give an example set of frequencies that would produce this case.

(b) Prove that if some character occurs with a frequency more than $2/5$, then there is guaranteed to be a codeword of length 1.

(c) If all characters occur with frequency less than $1/3$, then there is guaranteed to be no codeword of length 1.