CMSC 341
Treap Review Questions

1. Insert the following characters with their respective priorities (shown as ordered pairs) into an empty treap, then answer the questions below.

   (A, 17), (F, 22), (Q, 15), (K, 25), (R, 14), (B, 35),
   (X, 27), (M, 24), (L, 42), (T, 30)

   a. What is the height of the treap?
   b. What character is at the root of the treap?
   c. How many leaves does the treap have?
   d. How many levels does the treap have?
   e. Which character (if any) is the left child of M

2. Perform the following deletions from the treap created in question #1 above and then answer questions 1.a. thru 1.e. for the new treap
   a. Delete F
   b. Delete A
   c. Delete K

3. Some trees (such as binary heaps and up-trees) can be implemented in an array instead of using nodes and left/right child references. Can a treap be implemented as an array? If, describe how data would be stored and how the insert, delete and find operations would work. If not, explain why not.

4. Explain how treaps can be used to divide a finite set into two subsets such that all elements in one subset are less than a given key, K, and all values in the other subset are greater than K.

5. Treaps are useful because they are (almost) always balanced. Why is this so?

6. Treaps are often suggested as an alternative to Red-Black trees because they are simpler to implement and perform “reasonably well”, yet it is possible that a treap will degenerate into a linked-list whereas a RBT will not. Describe the circumstances under which a treap may become a linked-list.