**2)** (5 pts) ALG (Heaps)

Suppose we construct a minheap where each node contains a string, and we order our strings according to the following rules:

1. Given strings a and b, we say $a < b$ if a has fewer characters than b.
2. If two strings a and b have the same length, we say $a < b$ if a comes before b in alphabetical order.

Suppose we construct a minheap of strings using these rules. Furthermore, suppose all the strings in our minheap contain lowercase letters only (so, no punctuation, spaces, uppercase letters, and so on), and we do not allow any duplicate strings into the minheap.

Given two arbitrary strings in our minheap, x and y, can we safely say that if x is a prefix of y, then y must be in one of x's subtrees? Note that x may not be the root of the minheap. If so, explain why this must be the case. If not, draw a minheap of strings that very clearly shows this is not necessarily the case (and clearly label which string is x and which string is y in your counterexample).

Here's one such minheap, where x = "ef" and y = "efficacy":

```
            ab
          /      \
        cd        ef
       /
   efficacy
```
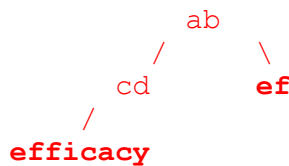
Notice that "ef" is a prefix of "efficacy", but "efficacy" is not in either subtree of "ef".

<u>Grading:</u>
+1 attempt to create a counter-example (0/5 if they say y must be a subtree of x)
+2 for creating a valid minheap using these rules
+2 for coming up with *x* and *y* that satisfy the problem (they should denote which nodes store x and y and x must be a prefix of y.)