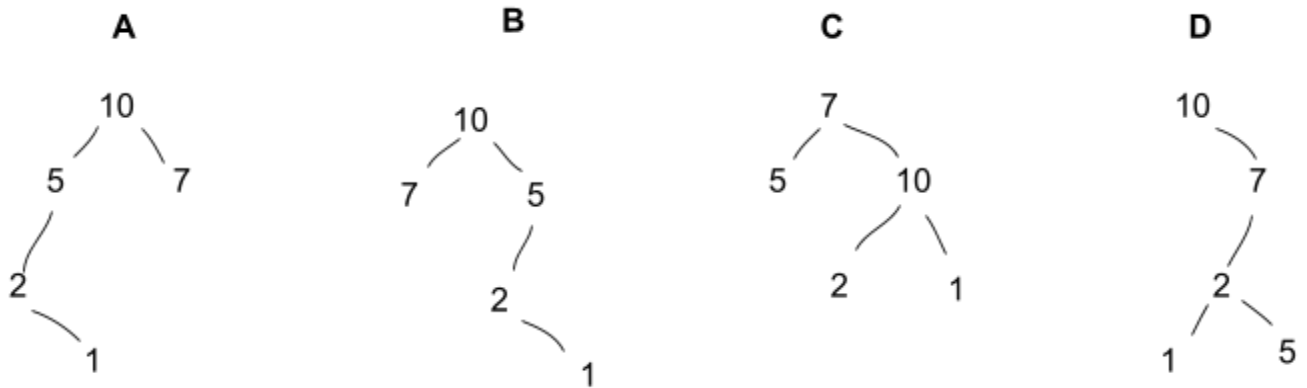


Heap: A binary tree in which the highest-priority item is at the root and both the left and right subtrees are also heaps

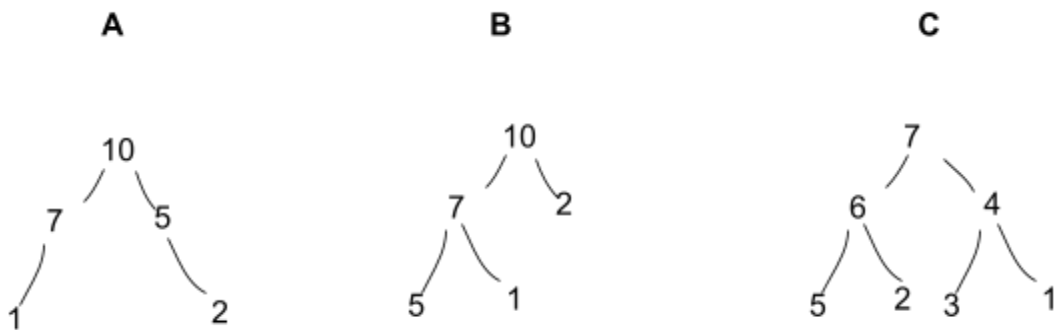
Exercise: Which of the following are heaps? Which are balanced (whether or not they are heaps)?



Define “balance”:

Goal: implement heaps with the run-times stated above.

Try it: insert 8 into each of the following trees, while maintaining requirements



Implementation: here's a binary-tree class in Python

```
class BinTree:
    def __init__(self, data, left=None, right=None):
        self.left = left
        self.right = right
        self.data = data

# a      unbalanced_tree = BinTree("a",
# \      left=None,
#  b      right=BinTree("b",
#   \      left=None,
#    c      right=BinTree("c",
#     /      left=BinTree("d"),
#    d      right=None)))
```

How to create an easier implementation?