Congratulations! You passed!
TO PASS 80% or higher

Keep Learning

GRADE
100%

Week 4 Challenge

LATEST SUBMISSION GRADE
100%

1. **Implement downHeap(Node *n) for a min heap implemented as an ordinary binary tree with an integer member variable "value" and left and right child pointers. Unlike the video lesson which implemented downHeap on an array implementation of a complete binary tree, the binary tree in this challenge problem is not stored as an array and is not necessarily complete; any node might have only a left child, only a right child, both, or neither.**

   The starter code below defines a class called "Node" that has two child pointers ("left" , "right") and an integer "value" member variable. There is also a constructor Node(int val) that initializes the children to nullptr and the node's value to val.

   Your job is to implement the procedure "downHeap(Node *n)" . The procedure should assume that n->left is the root of a min heap subtree (or nullptr) and the same for n->right. The value at Node *n (specifically n->value) might not be less than the values in its left subtree and in its right subtree, and so the tree with Node *n as its root might not be a min heap (even though its left subtree and right subtree are both min heaps). Your code should modify the tree rooted at Node *n so it is a min heap. You do not need to balance the tree or turn it into a complete tree. You only need to ensure that the tree satisfies the min heap property:

   For a min heap, it is okay for a node's value to be *equal* to one or both of its children, but the node's value must not be *greater* than either of its children.

   5 / 5 points

   Correct

```
Correct!
Original root value: 777
Your downHeap result:

 [[Note] As a temporary fix for properly reporting the
unit test's tree in the server output here, each newline
will begin with a "#" symbol, which you can replace with
a line break in your text editor for viewing. Padding spaces
are shown as "." instead. Thanks for your patience! We'll
improve this output soon.]

##8
#|
#|_ 17
#|..|
#|..|_ [null]
#|..|
#|..|_ 21
#|.....|
#|.....|_ 32
#|.....|..|
#|.....|..|_ [null]
#|.....|..|
#|.....|..|_ 777
#|.....|.....|
#|.....|.....|_ [null]
#|.....|.....|
#|.....|.....|_ [null]
#|.....|
#|.....|_ 23
#|........|
#|........|_ 27
#|........|..|
#|........|..|_ [null]
#|........|..|
#|........|..|_ [null]
#|........|
#|........|_ 31
#|..........|
#|..........|_ [null]
#|..........|
#|..........|_ [null]
#|
#|_ 16
#...|
#...|_ [null]
#...|
#...|_ 28
#......|
#......|_ [null]
#......|
#......|_ 37
#.........|
```

```
#.....|_ 43
#.....|..|
#.....|..|_ [null]
#.....|..|
#.....|..|_ [null]
#.....|
#.....|_ 49
#.....|
#.....|_ [null]
#.....|
#.....|_ [null]
```