POTD38.1. Problem of the Day #38

Download and Extract

An initial setup of files is provided to you via a shell script: Download potd-q38

Using a terminal, extract the initial files by running the shell script you just downloaded (you will need to navigate to the directory where you saved the file):

```
sh potd-q38.sh
```

Your files for this problem will be in the `potd-q38` directory.

The Problem

In today's POTD, you are going to implement one specific case of BTree removal: removing from leaf nodes. Complete the two functions: `underflows` and `rotateRight`. You don't need to worry about rotateLeft or removing from other nodes.

We handle the finding and removing element step for you. But there is one more problem. After removing one element, the number of elements inside the BTree node might become too empty! We call this an underflow.

You need to implement the function `underflows` to check if a BTreeNode with given `numElem` and `order` underflows. *Hint: What is the lower bound of the number of elements in an internal node?*

You also need to finish the function `rotateRight` with the actual rotation part. Since it is a leaf removal, you don't need to worry about children nodes.

Note: `removeFromLeaf` and `rotateRight` requires `underflows` to work. So make sure your `underflows` works before moving on.

Testing Your Code

In `main.cpp`, an exmaple BTree has been provided to help you test your code:

```
               | 40 |
             /          \
   | 10 | 20 | 30 |    | 50 | 60 |
```

Result:

```
        |40|
|10|20|30|    |50|60|

removing 50...
Does the node underflow? Yes!
Is the removal successful? Yes!

        |30|
|10|20|        |40|60|
```

Upload Solution

Drop files here or click to upload.

POTD 38

Total points:	0/1
Score:	0%

Question

Value:	1
History:	
Awarded points:	0/1

Report an error in this question

Previous question

Next question

Only the files listed below will be accepted—others will be ignored.

Files

○ BTreeNode.cpp
not uploaded

Save & Grade Save only