POTD17.1. Problem of the Day #17

Download and Extract

An initial setup of files is provided to you via a shell script: Download potd-q17

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-q17.sh
```

Your files for this problem will be in the `potd-q17` directory.

The Problem

Complete the `sortList` function that accepts a `Node **` (the head of a linked list, passed by pointer). This function must sort the linked list **without** allocating any nodes and **without** changing what's in the `data_` member variables. In other words, you are to sort the list **only** by manipulating the pointers.

This sort does not need to be the most efficient sort possible, it just needs to be correct. A `main` function has been provided that exercises your `sortList` function.

We gave you the skeleton code for selection sort. Use it if you wish

Hint: if you are getting stuck, draw a detailed memory diagram.

Sample Output

```
Unsorted List:
Node 0: 2
Node 1: 4
Node 2: 1
Node 3: 5
Node 4: 3
# of Nodes before sortList: 5
Sorted List:
Node 0: 1
Node 1: 2
Node 2: 3
Node 3: 4
Node 4: 5
# of Nodes after sortList: 5
```

Understanding `Node**`

If you do not understand why a `Node**` is being passed in, read the following: A `Node **` is a `Node *` passed by pointer. What would happen if we just normally passed a `Node *head` representing the head of the linked list into the function? We could still alter the node that head points to. But what if we want to change the head pointer itself? E.g., what if we wanted to insert a new node into the front of the list? If we just called the function like this: `insertSorted(head, newNode)` where head is a `Node *`, then the memory address stored in the pointer is copied into the function. The function can change the pointer, but this will only change the copy of the pointer, not the pointer that lives outside the function. To change the address stored in the pointer we need to pass by pointer. The function call would then look like: `insertNode(&head, Node)`. We also could have passed the pointer by reference, though we would have to change the function signature to be: `insertSorted(Node * & head, Node * n)`.

Upload Solution

Drop files here or click to upload.

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

POTD 17

Total points:	0/1
Score:	0%

Question

Value:	1
History:	
Awarded points:	0/1

Report an error in this question

Previous question

Next question

Files

○ Node.cpp

not uploaded

Save & Grade Save only