CS 15 Lab 8: Heaps

Introduction

In this lab you will implement the `min(), removeMin(), buildHeap(), downHeap(), and` ∼`Heap()` functions for a heap (but not `upHeap()`, unless you want to). The heap functions will be used by a sorting algorithm in `main()`. The heap is a min-heap and is implemented with an array. The heap is a heap of strings and, when sorted, will print a private message.

We will discuss a better way to do this sorting algorithm in class — to do it properly, it requires private access to the heap and destroys the data structure, but is more space efficient. In this lab, we're building a general heap. If you were creating the heap solely for sorting, you would probably do the sort in place.

The `remove` function is the same function as discussed in class: the root value will be returned, and the value in the rightmost node of the last level of the tree is copied to the root and down heaping begins from the root. The heap size then shrinks by one.

`buildHeap()` will start halfway through the list (array) and repeatedly calls down heap on the first half of the list. It is used by a non-default constructor which takes in a list of strings and builds a heap from the list.

`downHeap()` repeatedly swaps a node with its smallest child until the heap property is restored: all children are bigger than their parent.

```
std::string Heap::removeMin();
void        Heap::buildHeap();
void        Heap::downHeap(int location);
```

Each of these functions is in the class `Heap`. `buildHeap()` and `downHeap()` are `private` functions, while `removeMin()` is `public`. The `heapsort` function is given in `main.cpp`, it uses the heap constructor to build a heap, passing in a list of strings to sort. It then repeatedly calls `removeMin()`, storing each element into a temporary array. The temporary array is sorted

and is then copied back to the original array.

Getting Started

First make a directory for this lab, change into that directory, and copy the files from `/comp/15m1/files/lab08`. If you want to quickly determine where you need to write code, type:

```
grep -n TODO *
```

The grep program is a powerful search tool; when you have some time, read up on it!

Look in the `Makefile`. You'll see the name of the executable program it builds.

Submitting your work

Submit your work as discussed in class.