CMPT 225 D2
Fall 2020
T.Shermer

Assignment 4
Heapsort

Due Nov 6 at 23:59

You are to write an entry class and a heap class, any associated exception classes, and a subroutine to do heapsort. There will also be some code for testing this implementation.

The entry class should be called **Entry**, and it should hold a key (integer) and data (a string). It should have a method called **random**(), which gives it a random key between 0 and 99, inclusive, and a random data string of three lowercase letters. It should have getters for the key and the data. It should have a toString method which converts it to a string like so: (14, drn).

The principal class will be called **Heap,** and it should hold Entry objects. It should be a **max**-heap, so it will have operations:

```
type    name

Entry&  removeMax()          // returns the maximum entry in the heap, and deletes it.
void    insert(Entry& e)     // inserts the entry e into the heap.
int     size()               // number of elements in Heap
bool    empty()              // is size = 0?
void    make(Entry[] entries, int n)    // does the linear bottom-up makeHeap operation
                                        // to make the heap have all of the n entries in the
                                        // array. Throws away anything already in the heap.
```

It should also have a constructor and a destructor. You may use an array implementation of the heap or a linked (pointer) implementation. If you use an array, you must double it when it reaches capacity.

Note that the functions above (in both classes) are given approximately. Add references (&) where necessary or sensible, and add **const** to arguments and functions if they are const. I also haven't defined all the data members that the classes should have.

In your main file, write two subroutines heapsort1(entries, n) and heapsort2(entries, n), each of which returns the entries array with the n elements in it sorted from smallest key to largest key. Use a Heap to do this. In heapsort1, use insert() to build the heap, and in heapsort2, use make() to build the heap.

The tests to run are as follows: (1) create an array of 15 random entries, print it out, apply heapsort1, and print it again. (2) do the same with heapsort2. (3) same as (1) but with 31 entries. (4) same as (2) but with 31 entries. Format the printing nicely.

Entry and Heap should be in separate files (.cpp and/or .h) named with the class name. (Exception classes for the Heap may be in the Heap's file(s).) Your main should be in its own file as well.

You will be judged on correctness of your code and on code style, so don't forget to keep your code clean as you develop it! (Or at the very least, clean it up before submission. We don't want to see untidy code.)