COP 3503 - Programming Assignment # 5

Heaps

Assigned: March 25, 2014 (Tuesday)

Due: April 8, 2014 (Tuesday) at 11:55 PM WebCourses time

Implement an array-based max heap data structure and algorithms for (1) sorting a list of values using heapsort, (2) building a heap from a list of values using the bottom-up algorithm, (3) inserting a new value into a heap and (4) deleting the maximum value from a heap. This data structure and its related algorithms are described in section 6.4 of your textbook.

Input and Output

The input file **heapops.txt** specifies a list of heap operations, each on a separate line. An array is loaded using the **load** operation and is transformed using the **build-heap**, **heapsort**, **insert**, and **delete-max** operations. The current array is printed using the **print** operation. A **load** operation is guaranteed to be the first line of the input file.

- **load** *list-of-integers*
  - Example: `load 3 4 1 2 9 10 13 5`
  - Description: Constructs an array containing the specified (distinct) integers. A **load** operation replaces the current array which was loaded with a previous **load** operation. This array is not guaranteed to be a heap. The **build-heap** operation is used to transform this array into a heap. (**load** produces no output.)
- **print**
  - Description: Outputs the contents of the current array. This is the last array loaded with the **load** operation, which may have been modified by any **build-heap**, **heapsort**, **insert**, and **delete-max** operations that followed that **load** operation. Prints "(empty)" for an empty array. (An example output is *3 4 1 2 9 10 13 5* on a separate line.)
- **build-heap**
  - Description: Constructs a heap from the current array using the bottom-up algorithm given on page 229 of your textbook. This operation is guaranteed to be called before any **delete-max**, **insert** or **heapsort** operations. (**build-heap** produces no output.)
- **delete-max**
  - Description: Deletes the maximum value from the heap using the algorithm given on page 231 of your textbook. (**delete-max** produces no output.)
- **insert** *integer*
  - Example: `insert 3`
  - Description: Inserts a value into the heap using the algorithm given on page 230 of your textbook. (**insert** produces no output.)
- **heapsort**
  - Description: Sorts the values of the heap using the heapsort algorithm given on pages 231 – 232 of your textbook. Prints the values of the heap in decreasing order. After running **heapsort**, the heap becomes empty. (An example output is *13 10 9 5 4 3 2 1* on a separate line.)

Sample input

Please design your own test cases and test your algorithms thoroughly. Your Program producing correct output for the sample input may not be sufficient to verify that everything is working correctly.

```
load 2 9 7 6 5 8
print
build-heap
print
delete-max
print
heapsort
insert 3
print
load 9
build-heap
insert 2
print
delete-max
delete-max
print
```

Sample output

```
2 9 7 6 5 8
9 6 8 2 5 7
8 6 7 2 5
8 7 6 5 2
3
9 2
(empty)
```

Restrictions on Source Code

- Submit a file named **Heap.java** which defines a *public class Heap* which defines a *public static void main(String[])* method. This method Reads the input file "heapops.txt" and produces the desired output to System.out. You may submit other Java files as well.
- Do not put any of your your Java files in a package. Omit the package statement to leave them in the default package.
- Your program must compile using Java 7.0 or later. It's okay to develop your program using the IDE of your choice, although Eclipse is recommended. Your program should include a header comment with the following information: your name, course number, section number, assignment title, and date.

Deliverables

You must submit Heap.java and any additional Java files to WebCourses by 11:55 PM on Tuesday, April 8, 2014. You must send your source files as an attachment using the "Add Attachments" button. Assignments that are typed into the submission box will not be accepted. Assignments that are 1 day late are deducted 25% of the points received. Assignments more than 1 day late are not accepted. Programs that do not compile will receive no credit.