Homework Assignment #7 – Heaps (and Priority Queues)
by Evan Olds
Cpt S 223 – Advanced Data Structures

Submission Instructions:

Submit source code (zipped) to Angel BEFORE the due date/time. If the Angel submission is not working, then submit to TA via email BEFORE the due date/time.

Optional: Include a readme.txt file in the zip with any relevant information that you want the grader to be aware of.

Assignment Instructions:

Open the Visual Studio 2013 project included with this assignment. You will need to implement a priority queue in two ways. Recall that a priority queue always removes the maximum value. We discussed in class how a heap and a linked-list can both be used to implement a priority queue. In this assignment you must demonstrate this by finishing the following:

1. (4 points) Implement the PQ_LinkedList class defined in the PQ_LinkedList.h header file.
    - You may use a singly or doubly-linked list.
    - Do NOT use an array.
    - Do NOT use any containers from the standard library, such as vector or queue.
    - Keep the maximum value at the front/top of your queue so that removing it happens in O(1) time.
    - Implement your Add function such that it inserts the new item in the proper spot. The list should be sorted in descending order at all times.
2. (5 points) Implement the PQ_Heap class defined in the PQ_Heap.h header file.
    - Use either an array or a std::vector object to store the heap data. No other options will be accepted.
    - Properly implement the heap so that it uses the k-value passed to the constructor and builds a k-ary heap. Hard-coded binary heap implementations will not receive full credit. You must support ternary (k=3) and others for any value of k >= 2.
    - You can add helper functions such as sift up and sift down. Make them private members of the class.

Both of these classes inherit from an abstract base class defined in PQBase.h. Look at the declaration of this class before you start the implementation of either of the two classes. This way you'll know what functions you'll have to implement. The code will not compile until you implement all virtual functions in the inheriting classes.

If you've implemented both classes and it still doesn't compile, fix the implementation of those two classes. Do NOT change code in the HW7_Main.cpp file nor the PQBase.h file. If your code doesn't compile it’s not a problem with either of these two files. It’s a problem in one of your implementations of the inheriting classes.

The last 1 point of the assignment is for having clean, well commented code. If your code has too few comments or is poorly designed (e.g. not using the template types correctly or having “hacky” implementations that only work for certain data types) then you may not be awarded this point.

10 points total