Assignment 3 Report

For this assignment, my design consisted of only the PQ.h file and AvlTree.h implementations along with the PQdemo.cpp test file and makefile. I decided to put all the heap functions in PQ.h along with all the methods that were given in PQdeclare.h. To create the priority queue, I used a vector of structures, with each structure containing an integer type for the priority and a pointer to the AVL nodes in AvlTree.h. The functions that I added to PQ.h on top of everything from PQdeclare.h was percolate up, percolate down, build heap, and two swap functions, one for integers and the other for pointers to AVL nodes. In the AvlTree.h file, I added a friend class to allow PQ.h to access private functions and members and also added an index finder to return back an integer index to the update priority function in PQ.h. In both header files, I used the type name “ID” and added an index declaration in the AVL node structure and also set the id numbers to type “ID”. I was able to connect the heap pointers to the corresponding nodes by returning a void pointer from the insert function in AvlTree.h and type casting them to be able to access the members of the AVL nodes directly from my PQ.h implementations. I used “typename AvlTree<ID>::AvlNode*” to create the pointers. When swapping the priorities in the heap, I maintained the order invariant in the AVL tree by swapping the pointers and the indexes. Most importantly, the most crucial part that allowed me to finish this assignment was type casting the void pointers returned from the insert function in AvlTree.h.