Homework Assignment #6 – 2-3-4 Tree Class Implementation

by Evan Olds

Cpt S 223 – Fall 2013

**Submission Instructions:**

Submit source code (zipped) to Angel BEFORE the due date/time. If the Angel submission is not working, then submit to TA (shajiami@eecs.wsu.edu) via email BEFORE the due date/time. No late submissions will be accepted. “Angel wasn’t working” is never an excuse. Email the assignment if Angel isn’t working and make sure you email it before the due date/time or else you will be given a 0.

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

**Reminder of policy to request instruction changes:**

If you see some way that you could demonstrate the knowledge for this assignment by doing a different type of project, then contact me to discuss. Remember that I may reject your request and require you to do the assignment exactly as it’s written. But if you have a good idea of how you could alter the assignment to make it more useful to you, then email me with your proposal.

**Assignment Instructions:**

**Read all the instructions *carefully* before you write any code.**

Download the zip file from Angel and open the Visual Studio 2012 project included within it. Do not create a new project. Open the existing one from the zip. Complete the implementation of the hw6\_234tree class functions in the file hw6\_234tree.h. Do not modify any other files in the project.

1. Implement basic functions in the MyString class (10 points):

* T\* Find(int key) – Finds an object with the specified key in the tree. Use the GetKey function of the objects in the tree to get key values. Return the object that contains the key.
* bool Add(T& toCopyAndAdd) – Adds the object to the tree. Returns false if an object with the same key already exists in the tree. If it doesn’t already exist, then the object is COPIED first, then added to the tree. Use the copy constructor of the T object to copy it.

You must implement these 2 functions correctly for full credit on this assignment. Your trees should EXACTLY match those in the output files when using the test input files. There are 2 input files, marked as easy and medium, that are relevant for the addition functionality. It is recommended that you create additional input files to further test stability, as the grading process is likely to use test files different from the ones given in the project .zip file.

2. Implement the Remove function for up to 10 points extra credit:

Since it is significantly more challenging, the remove function is optional for extra points. You can get up to 10 points if you implement it correctly. You must correctly fuse nodes and handle all fusion cases:

1. Take from left child
2. Take from right child
3. Take from parent
4. Fuse root if the above three are not options