24-783 Problem Set 4

The following instructions assumes that your directory structure is:

```
(User Directory)
        eng_comp
                src
                        CMakeLists.txt    (Top-level)
                        public
                        (Your Andrew ID)
```

Also in the instruction, a forward slash (/) is used as the directory separator. In Windows command prompt, you need to use a back slash (\) instead.

You can use a different directory structure that is comfortable for you. However, in the SVN server, your project directory must be under:

https://ramennoodle/svn/teaching/24783_S16/students/(Your Andrew Id)

And, the project-directory name must comply with the instruction in each problem. Also TARGET_NAME of CMakeLists.txt must also comply with the instruction. Otherwise, the grading script cannot find your submission.

Test your source files with one of the compiler servers:

http://freefood1.me.cmu.edu:24780
http://freefood2.me.cmu.edu:24780

Make sure the server does not show any errors (red-lines) before submitting.

Submission is done by committing to the course SVN server (ramennoodle). You can commit as many times as you want. Teaching assistants will check out your submitted files at the deadline, and the grade will be given to what you stored in the repository at the time of check out. If you haven't submitted your program by then, late policy will apply.

<u>Deadline: 03/06 (Mon) 23:59</u>

Preparation: Set up CMake projects for bitmap and utility libraries and ps4 executable

You first create projects for the two problem sets

In the command line window change directory to:

~/24783/src/*yourAndrewId*

1. Update the course_files repository by typing:
   svn update ~/24783/src/course_files
2. Use "svn copy" to copy the bitmap-class and hash-table class source files explained in class to this directory. The files are in course_files. The command you type is:
   ```
   svn copy ~/24783/src/course_files/utility/bintree.h utility
   ```
   "utility" directory is from the previous problem set.
   (By the way, don't overwrite files you wrote for PS3. Your PS3 haven't been graded yet. Just copy bintree.h)
3. Create a sub-directory called:
   ps4
   and then, inside ps3 create sub-directories:
   ps4_1
   ps4_2
   <u>File/Directory names are case sensitive</u>. Use underscore. NOT hypen.
   Also the directory structure under your SVN directory is important. The grading script expects that the directory structure under your SVN directory is:
   ```
   ps4
       ps4_1
       ps4_2
   utility
   ```
   (By the way, don't delete ps3 and simplebitmap directories.)
4. Copy main.cpp of the binary-tree visualizer:
   ~/24783/src/course_files/bintreevis/main.cpp
   to ps4_1 and ps4_2 directories.
5. Write CMakeLists.txt for ps4_1 and ps4_2 sub-directories. The project is a graphical application. Therefore use MACOSX_BUNDLE keyword. The project name must be "ps4_2". Case sensitive and use underscore. Do not use hyphen. It must link "fslazywindow", "ysbitmapfont", and "utility" libraries.
6. Modify top-level CMakeLists.txt so that your build tree includes utility, and ps4/ps4_1 and ps4/ps4_2 sub-directories.
7. Run CMake, compile, and run ps4_1 and ps4_2 executables.
8. Add all the files you created to the control of svn. Svn-copied files are already under SVN's control, and you don't have to add them.
9. Commit to the SVN server.
10. Optional: Check out your directory in a different location and see if all the files are in the server.

PS4-1 Binary-Tree Re-balancing (60 points)

In PS4-1, you implement the binary-tree re-balancing algorithm presented by Quentin F. Stout and Bette L. Warren in 1986. (http://web.eecs.umich.edu/~qstout/pap/CACM86.pdf)

(1) Download and read the paper.
(2) Implement right rotation function in the BinaryTree class in:
    ~/24783/src/*yourAndrewId* utility/bintree.h
(3) In main.cpp, the user needs to be able to apply right-rotation by R key while the mouse cursor is on the target node.
(4) Implement TreeToVine function in the BinaryTree class.
(5) In main.cpp the user needs to be able to apply tree-to-vine by V key.
(6) Implement Compress and VineToTree function.
(7) In main.cpp the user needs to be able to apply vine-to-tree by T key.
(8) In main.cpp when the user presses the SPACE key, insert a new random number (between 0 and 99) to the binary tree.

If you successfully implement the four member functions and modify main.cpp accordingly, you must be able to re-balance the tree by pressing V and T keys.

PS4-2 AVL-tree (40 points)

In PS4-2, you implement a type of self-balancing binary-tree called AVL-tree. Read the explanation in the lecture note for more details about the algorithm.

(1) In BinaryTree class, add a flag (bool) called autoRebalance. Make it a public member variable.
(2) In the constructor, set autoRebalance to false so that by default auto-rebalancing is off.
(3) Write a protected member function called Rebalance, which takes a node pointer (or can be a handle) and applies the re-balancing algorithm of the AVL-tree. This function must check and re-balance the given node and all the up-stream nodes.
(4) In Insert function, if autoRebalance flag is true apply Rebalance at the appropriate location.
(5) In Delete function, if autoRebalance flag is true apply Rebalance at the appropriate location.
(6) In ps4_2/main.cpp, set autoRebalance flag to true in Initialize function BEFORE ADDING NUMBERS.
(7) In ps4_2/main.cpp when the user presses the SPACE key, insert a new random number (between 0 and 99) to the binary tree.

AVL-tree Performance Competition

We measure time that your AVL-tree class takes for adding 10,000,000 random numbers, and then deletes all even-numbers. Top-3 submissions will get 20 bonus points. Next 3 will get 10 bonus points.

If your code beats my AVL-tree class that I'm using for my research work, you'll get 3% bonus toward the final grade. By the way, my code takes 11.50 sec for insertion, and 1.44 sec for deletion, 12.93 sec in total. It is not the fastest code in the world. Some parts are written in a conservative fashion. You have a good chance of beating it!