COP4530 – Data Structures, Algorithms and Generic Programming
Recitation: 8
Date: 21th and 24th October, 2011

Objective:

1. Discuss Assignment 4
2. Discuss last quiz
3. Take the optional quiz

Assignment 4 discussion:

Part I (Implementing MRBST)

- First create a generic MRBST class in a header file **MRBST.h**.
- An MRBST class is a self restructuring BST. The restructuring property is given as follows:

  Restructuring is only done in the ***search*** operation.
  If the node being searched does not have a right child do nothing.
  Otherwise, rotate the node with its right child. (case 4)

- Moreover you have to implement the following operations:

void push(const T &): Inserts a node if it does not exist (Please note that here the node is created inside the push function and it is called only with the data as an argument)

bool search(const T &): Search for a node, returns true if found. Searches with the data value again. The restructuring must be implemented in this function.

void PrintPreorder(): Prints the preorder traversal of the tree.

A default constructor: Constructor without any arguments. Creates an empty tree

A Destructor: Free the allocated memory if any.

Please make sure the that your method names match with the names given in the assignment page. Otherwise your code will fail to compile when we try it.

Part II (Test MRBST)

- Write some code in a file called ***test.cpp*** showing that your MRBST class is working properly.

You might insert a few nodes and then search and write some code to check if the restructuring function is working correctly or not and test the preorder traversal.

Part III (Compare Containers)

You have to create an executable called *compare-containers* that does the following :

- Stores all the words from ***/usr/share/dict/words*** into 3 containers viz.

    1. MRBST
    2. STL List
    3. STL Set

- When a word is searched for, it checks all three containers and if the word is found it prints out the search time for each of the containers (You might use ***gettimeofday*** function to get the search time)

Part IV (result.txt)

This file should contain the following:

1. The description of your test cases in *test.cpp*. You should mention what properties you tested, how you created the test case and whether it passed the tests.
2. The relative performance of MRBST with the other two containers from STL. Mention in which situation MRBST is better than the other two and justify your results with the search time results.
3. Also mention if the MRBST had rotated searched words one level up, then would its performance have been better for this application.