Jean-Marc Boullianne
jboullia@u.rochester.edu
Assignment: 05
CSC 254: Fall 2015

Assignment #5: Templates

Summary:

The purpose of this assignment was to implement a generic ordered set abstraction. It was written in C++ utilizing the template functionality of the language. We also had to convert certain set operations to O(n) time complexity (linear time), and implement explicit comparators.

How I solved the problem:

I solved this problem in 3 steps:

1. Converted O(n^2) set operations to O(n) time complexity. I did this by utilizing the iterator class that was pre-implemented inside the given source code.
2. Converted the “oset” class from an int only based set to a generic set. I did this by utilizing the template functionality of C++.
3. Added explicit comparators for the generic ordered set. I add an argument for the “oset” constructors which was a bool function that pointed to a comparator. When a set operation was performed on two same type sets which had different comparators, I used the given O(n^2) set operations.

I created two helper functions to assist me with making the set operation O(n).

- insertval(node* p, T v)
- deleteval(node* p)

These two functions were monumentally useful when iterating over both set’s because they saved me the trouble of searching through the set to find a value, I knew I already had, because I was using the iterators. From there it was trivial to convert the O(n^2) set operations to O(n).

How To Run:

1. Open oset.cc and choose which one of the test cases you would like to run.
2. compile source code & run

Files Included:

- ➔ oset.h : Python program to cross index an executable file that’s input
- ➔ oset.cc : Contains many test cases used to test my program. Also shows examples of how to use “oset” class.
- ➔ SampleOutput.txt : A snippet of a sample run from my program executing