

CSCI 230 -- Lab 6

Sets and Union/Find Structures

Due: _____

Feel free to discuss and help each other out, but does not imply that you can give away your code or your answers! Make sure to read all instructions before attempting this lab. You can work with a lab partner and submit one lab package for your group.

You must use an appropriate provided template from Canvas or my website (zeus.mtsac.edu/~tvo) and output "Author: Your Name(s)" for all your programs. If you are modifying an existing program, use "Modified by: Your Name(s)".

Lab question 1: Describe the *template method pattern*.

Lab question 2: What are some applications for the find/union partition structures?

Implement one of the exercises below:

- Use classes **Merge**, **UnionMerge**, **IntersectMerge**, and **SubtractMerge** from C++ book to perform basic set operations. Set up a driver to test the three set operations: union, intersection, and difference. There is no equivalent code in Java, but you can set up your own code using the four classes above as a guide.
- Implement the find/union partition structure as a list-based approach as discussed in class/book.
- **Must be done in C++ unless it is for extra credit.** Implement the find/union partition structure as a tree-based approach. In Java, use Partition.java and Position.java, add a driver to test it for n operations (make cluster/set, find, and union) and collect some data to confirm it would cost $n \log n$ for n operations. For C++, use Partition.java and Position.java as a guide to create your own code, add a driver to test it for n operations (make cluster/set, find, and union). You can ignore the validate operation as long as you refer to a valid position.

Extra Credit: Implement another exercise above.

Online Submission: Submit one PDF file via Canvas includes status, answers to lab questions, output and source code for all required programs.