CS 127: Data Structures **Winter 2006**
Handout 9: Problem Set #7: Project and Presentation *27 February 2006*

> *Never tell people how to do things. Tell them what to do and they will surprise you with their ingenuity.*
> — George Patton (1885–1945).

The final assignment of the term is a little different from previous assignments. We're going to devote the last two days of the term to new data structures that we haven't covered in this course, and, in a group, you're going to be presenting a data structure to your classmates.

Presentation format: Your group will give a twenty-minute presentation on your data structure in class on Wednesday, 8 March 2006 or Friday, 10 March 2006. I've assigned timeslots randomly. *I will be strict about time—we have to fit all the presentations into two class periods—so be prepared!* You're encouraged to split up the parts of the presentation so that you each do some of the talking.

Your goal is to explain what the data structure does (i.e., what ADT does it implement? how efficient are the operations?), and give the class an idea of how it's done. You won't have time to show us oodles of code, but you can show us some pseudocode and draw a few pictures (or prepare a slide/handout with pseudocode/pictures). You're also welcome to do a code demo if that will help us to understand the data structure. Feel free to use an applet (even if written by someone else) if it can help us visualize what you're talking about. You can use whatever resources you want to describe your data structure: props, blackboard and chalk, prepared slides, handouts, demos, or anything else that you want to do.

Report format: The written part of your project will have two components:

(1) A ∼2-page single-spaced typewritten document explaining in written form the same material as in the presentation. What problem does this data structure solve? How does it solve it? Diagrams are appreciated when they help with clarity. (They do not count towards the page count.)

(2) Pseudocode for each of your data structure's operations. Comment clearly! *Optionally,* you may implement your data structure and submit the code. Although an implementation is not required, it will be treated as a very positive part of your report. I will be much more lenient in evaluating a project with a working implementation than in evaluating one without.

I expect you to consult multiple sources: the textbook, other textbooks in the library—*Introduction to Algorithms* ("CLRS") by Cormen, Leiserson, Rivest, and Stein is a good place to try—Wikipedia, and other web pages that you may find. Be *sure* to cite your sources in your written report.

Evaluation: Your evaluation will be based equally on your presentation and your report. I'll be looking for clarity of presentation in addition to the correctness of technical details. The written reports will be graded using similar criteria, and they will also be evaluated on the basis of your use of the English language. You will be asked to fill out peer evaluations, and your grade will also depend on the contributions that you made to your team's effort. You are not required to all work on all aspects of the project, but you *are* all expected to do your fair share.

Groups: Here are your randomly assigned groups, presentation times, and the data structures:

(Wednesday) splay trees: Kyle Kingsbury, Rob Thomas, Kelson Zawack.
(Wednesday) treaps: Keith Carr-Lee, Andy Nieuwkoop, Ellen Tighe.
(Wednesday) union-find: Michael Amundson, Don Penman, David Selassie, Emma Turetsky. (Look up "union-find" in Wikipedia.)

(Friday) tries: Michael Feinberg, Tom Hagman, Locke Perkins, Jerad Phelps.
(Friday) skip lists: Gabe Hart, Jacob Hilty, Andrew Robb.