Planning Assessment for a Game-Like, Highly Reusable Data Structures Assignment

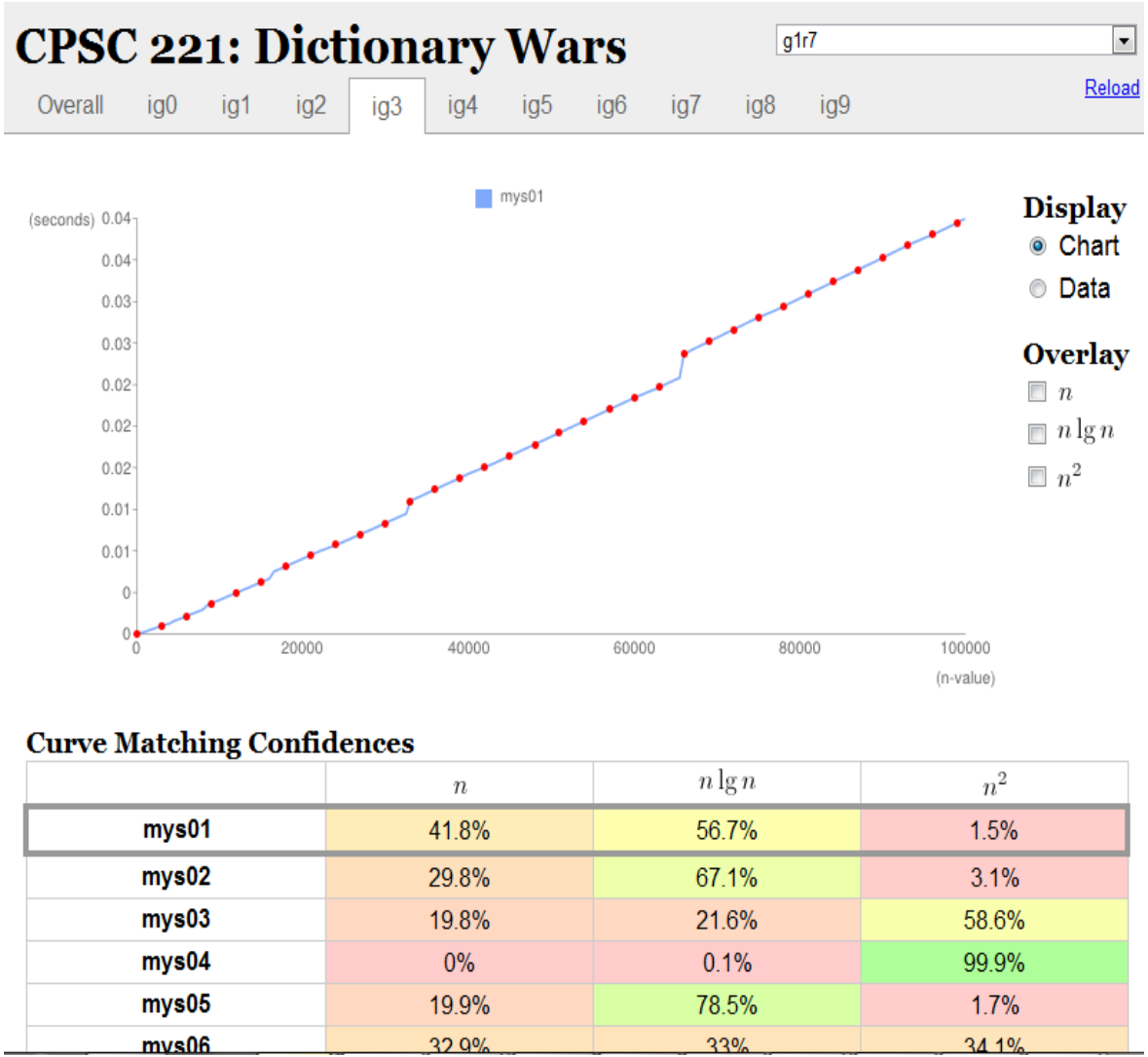
Steve Wolfman, CPSC

Students submit programs that use a data structure with an unknown implementation:

Student reasoning:

Students match up 6-10 possible implementations with "mystery" versions.

This student might hypothesize **mys01** is the array-based sorted list with resizing because of the small but increasing "jumps" in the graph.



	n	$n \lg n$	n^2
mys01	41.8%	56.7%	1.5%
mys02	29.8%	67.1%	3.1%
mys03	19.8%	21.6%	58.6%
mys04	0%	0.1%	99.9%
mys05	19.9%	78.5%	1.7%
mys06	32.9%	33%	34.1%

Automated reasoning:

The server runs and plots results from students' test cases. They're scored on "asymptotic" distinction.

Here, **mys01** is clearly not a quadratic curve, but **mys04** was.

Project Goals

Teach students to: Predict, measure, and judge the importance of the strengths and weaknesses of alternative abstract data type implementations.

Next steps: “Turn-key” automation of the project. Instructor makes a few high-level choices (which structures, what semantics, who to disseminate to). Each student receives a unique but related problem to solve. Students easily receive semi-automated feedback as they go.

Assessment

Providing 2-3 different questions related to the project on the exam (tomorrow). How should we assess student learning in the long run?

Roughly sketch the dictionaries' behaviour responding to the commands "INSERT 100000, INSERT 99999, ..., INSERT 3, INSERT 2, INSERT 1" on the axes below. The vertical axis should be the **total time** for all operations completed to that point (as in the programming project). Label and briefly explain key elements of the sketches (e.g., unusual properties, asymptotic behaviour, and relative runtimes between graphs).

(a) Unsorted LL:

Imagine the keys 1–4,000 have been inserted in random order … Give a strategy to distinguish each pair of dictionaries using only find and remove operations. … For full credit, your strategy should *absolutely unambiguously* distinguish the dictionaries.

[Setup explaining a company using a binary heap to solve a problem.] Unfortunately, after a ``cold reboot'' of the system … they suffer occasional delays during their first busy period after the reboot. Subsequent busy periods are fine until the next ``cold reboot''. Using your knowledge of binary heaps, hypothesize what is causing the trouble.