

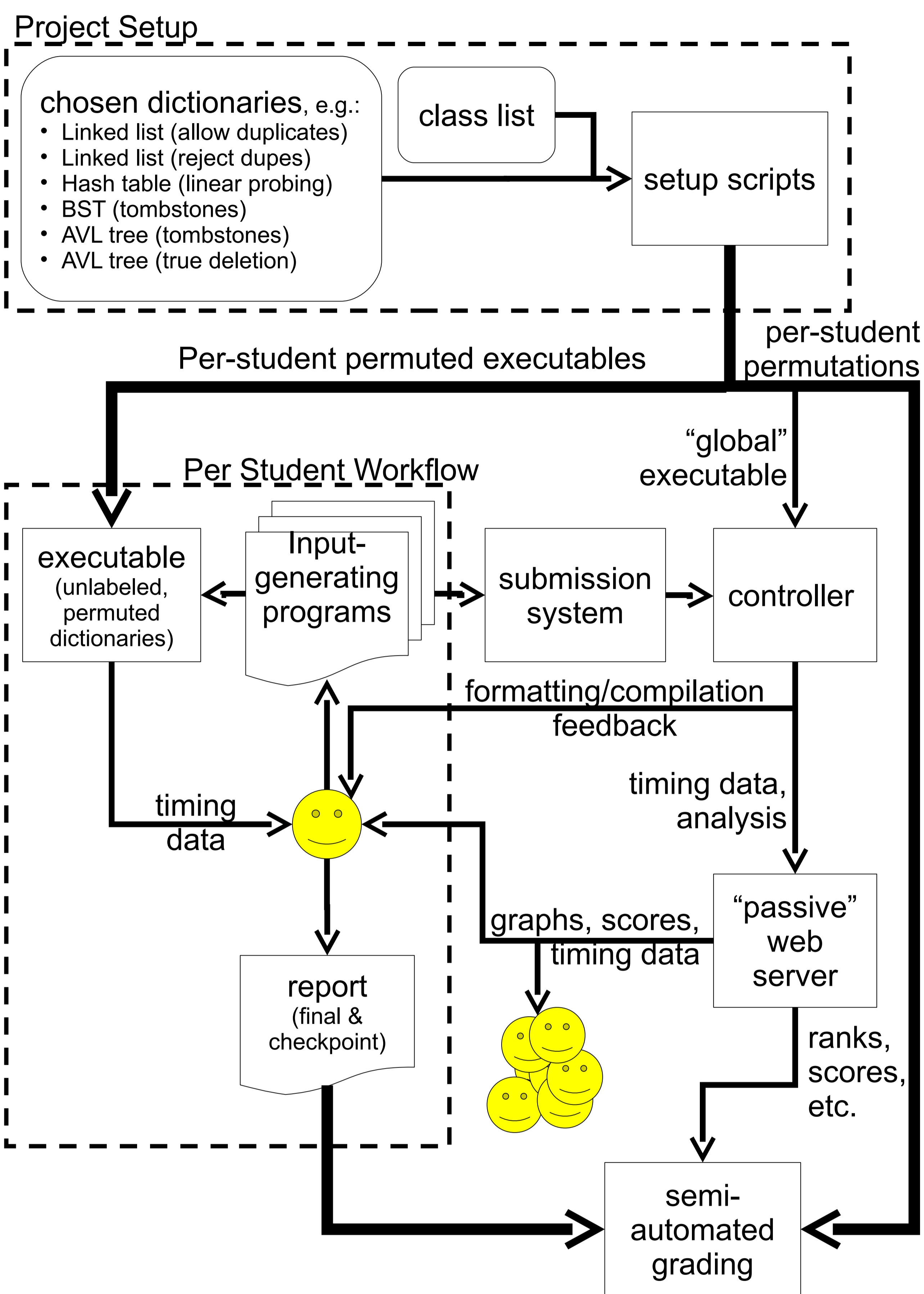
“Dictionary Wars”: An Inverted, Leaderboard-Driven Project for Learning Dictionary Data Structures

Kuba Karpierz*, Joel Kitching*, Brendan Shillingford*, Elizabeth Patitsas†, Steven A. Wolfman*

* University of British Columbia

† University of Toronto

Contact Authors: mr.karpierz@gmail.com, wolf@cs.ubc.ca

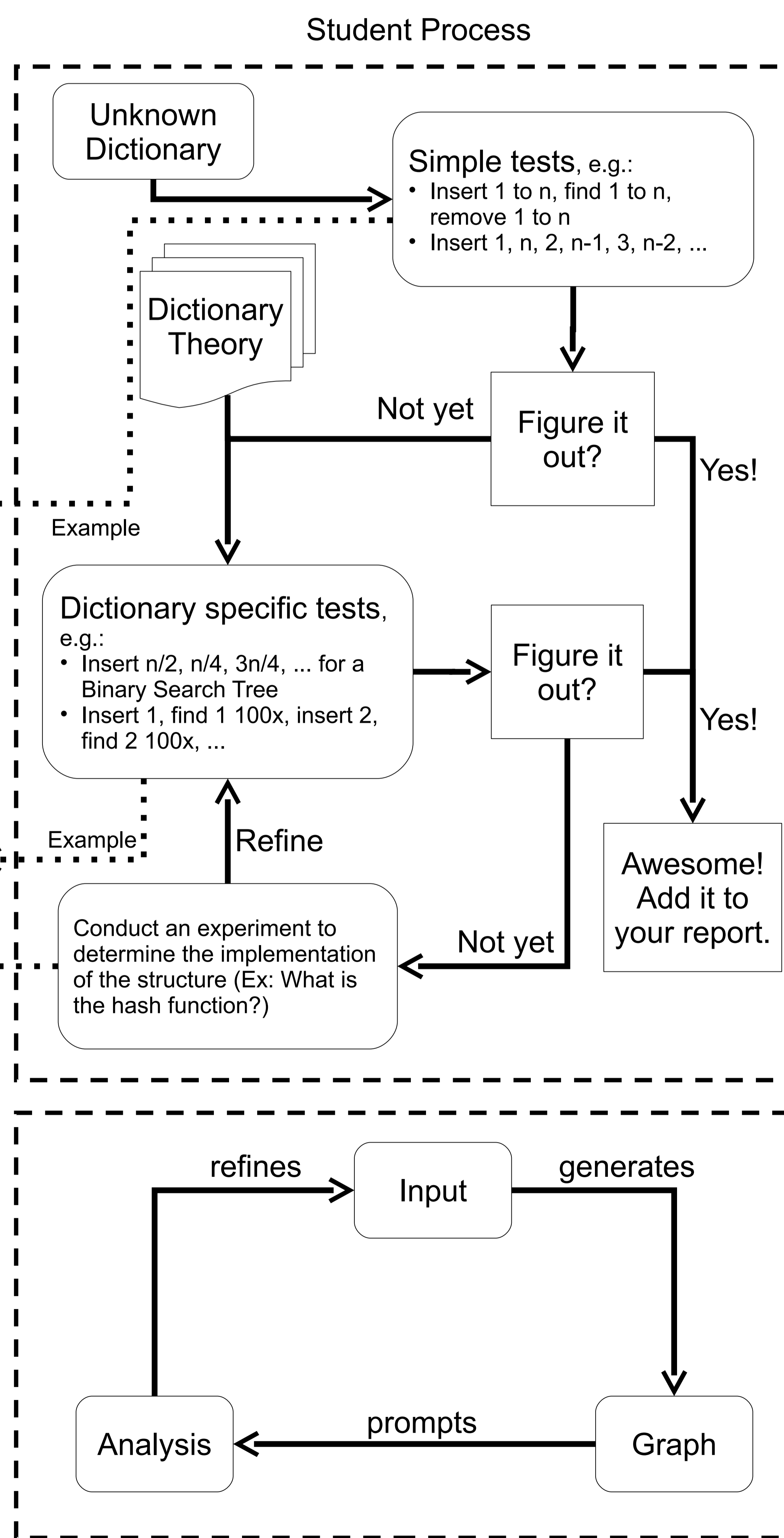
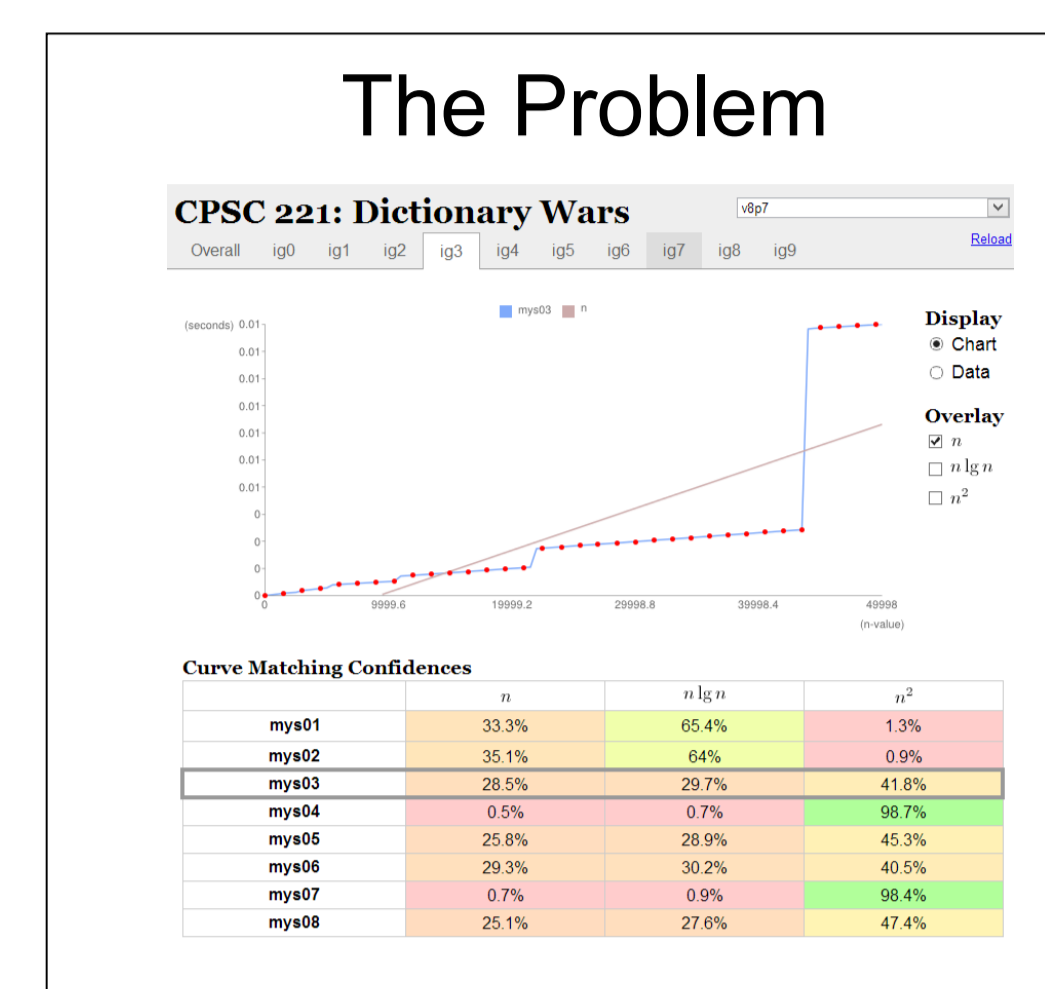
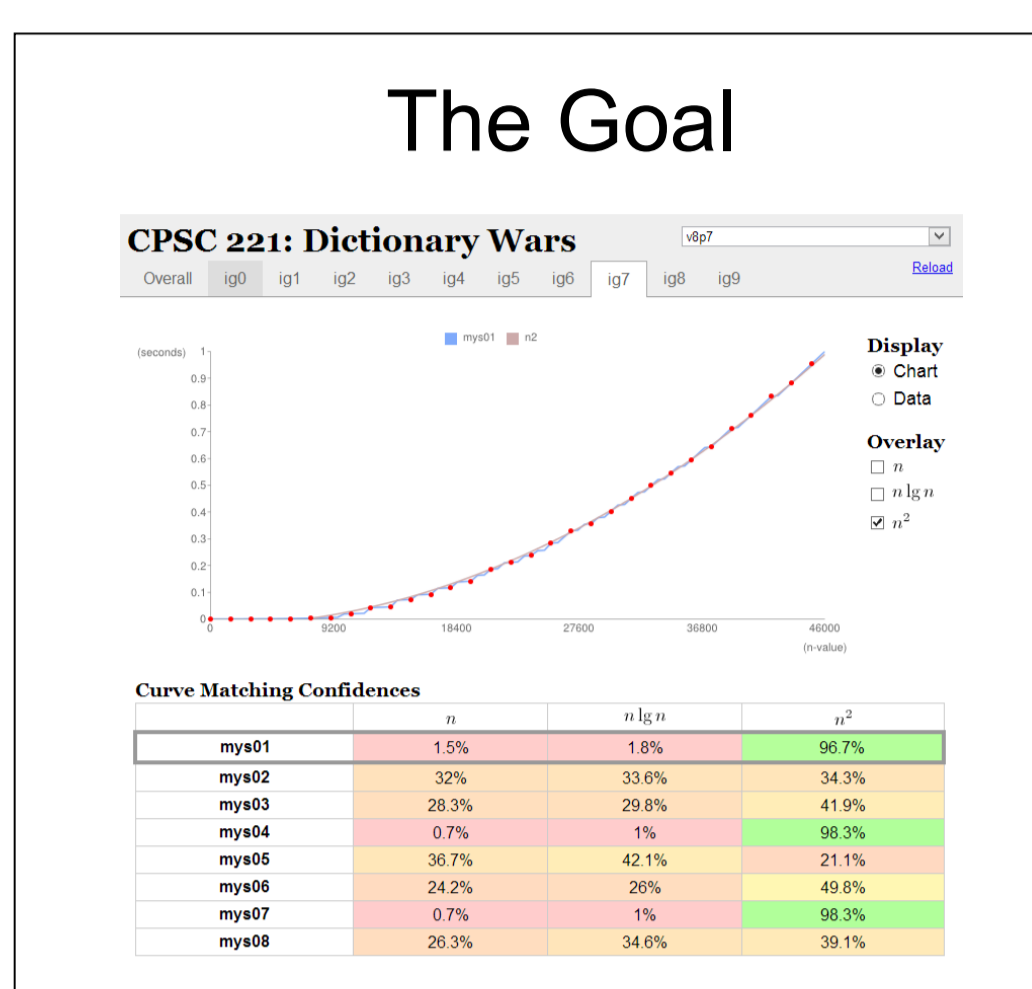
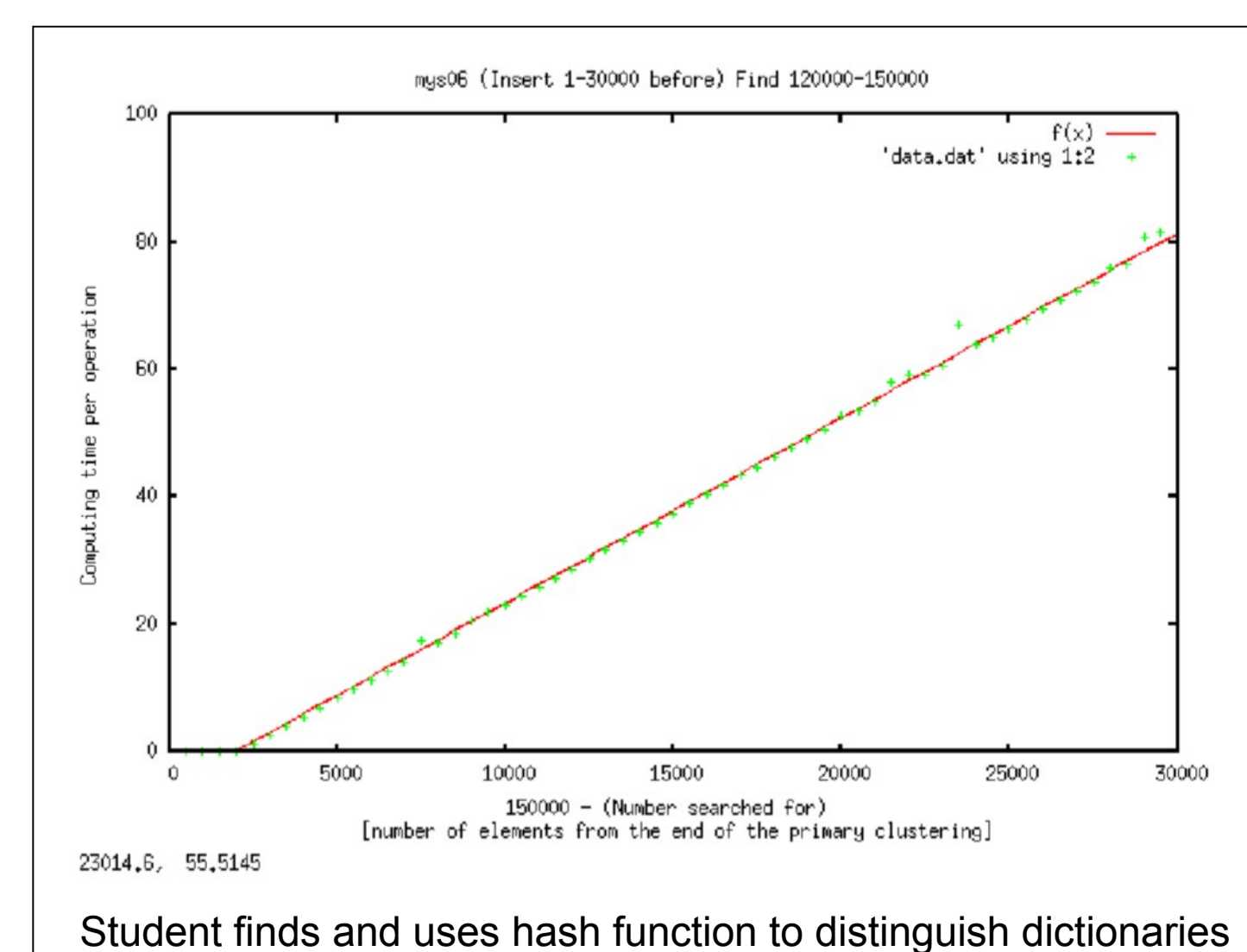
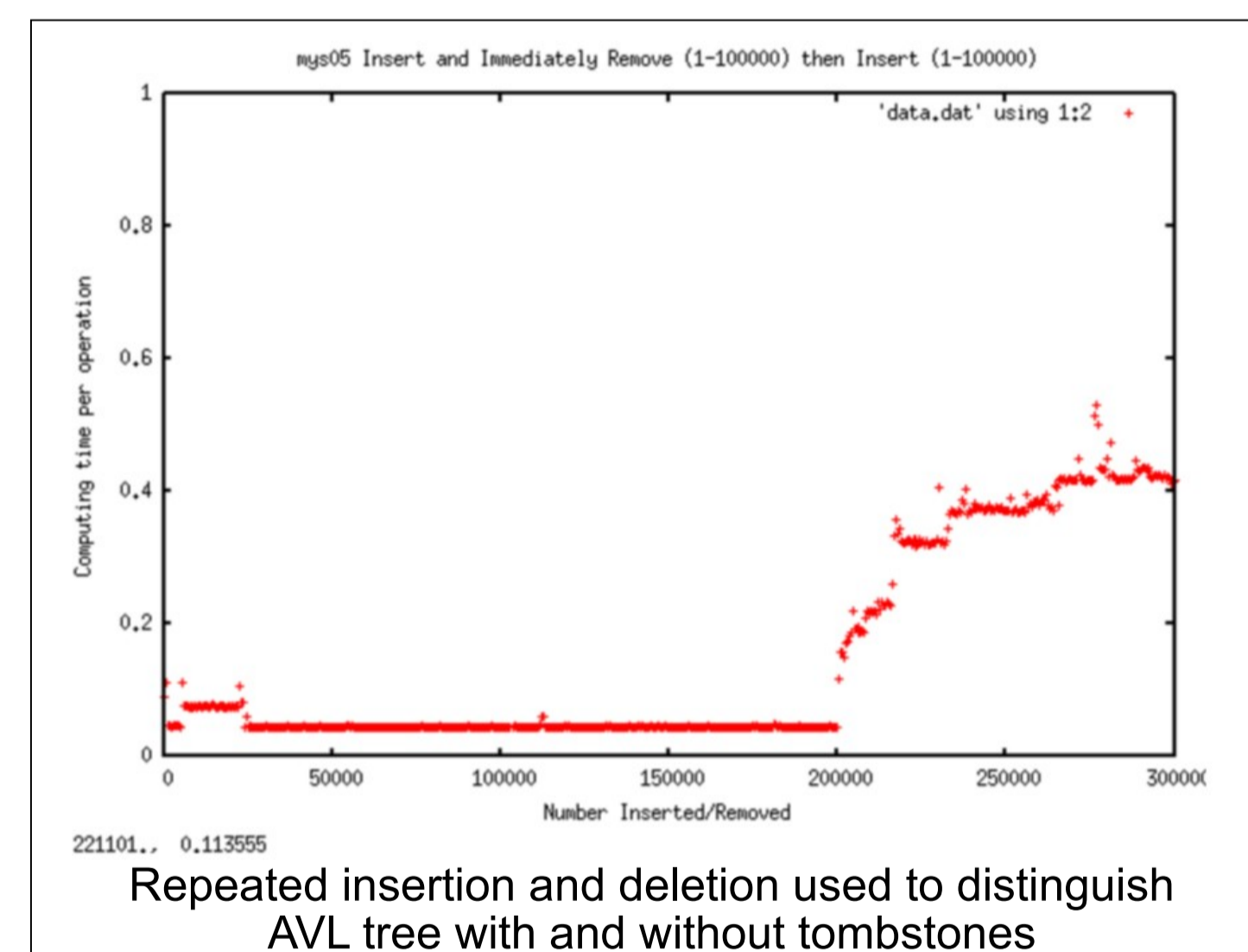
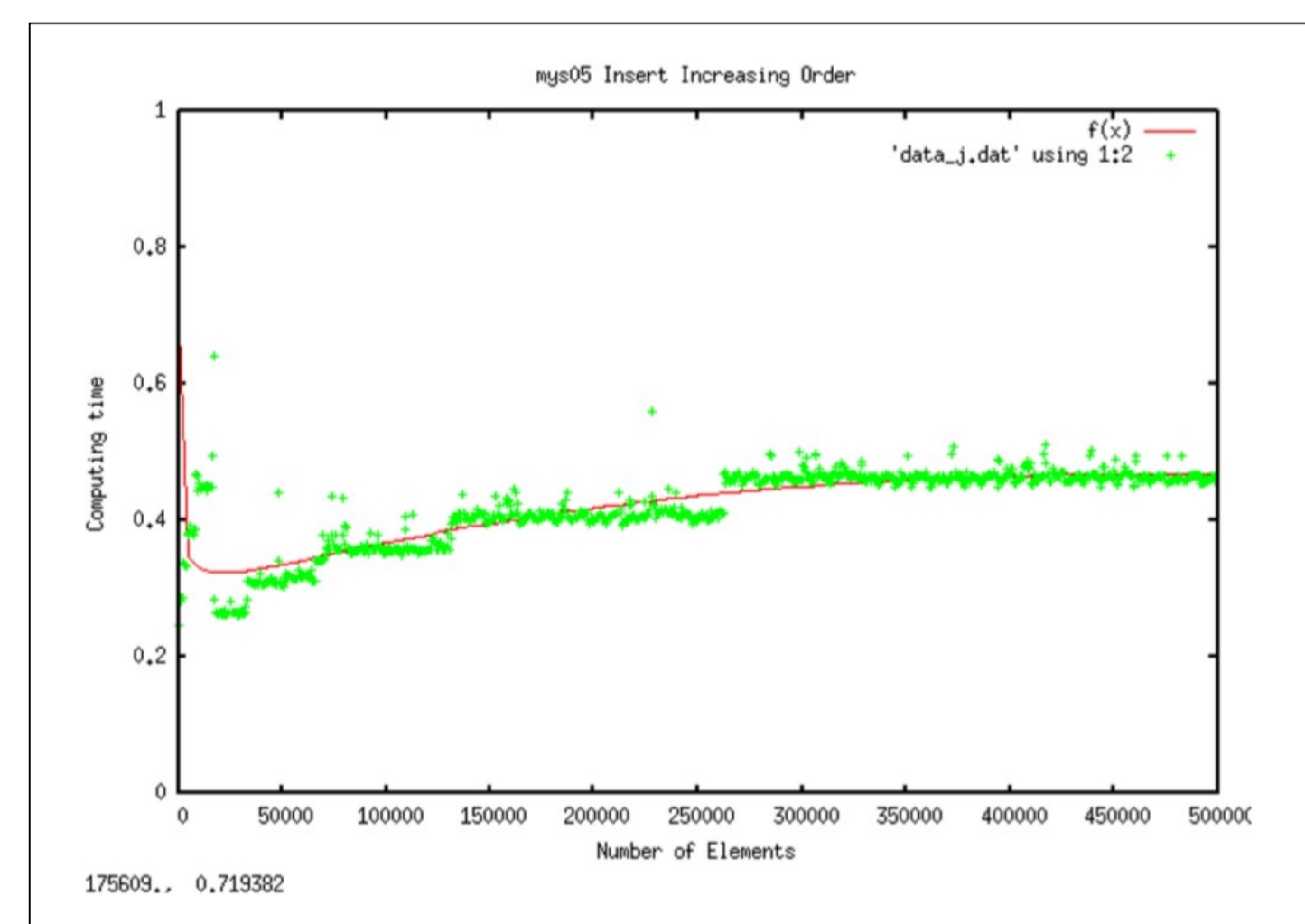


The Motivation

CPSC 221: Dictionary Wars

Overall User Rankings

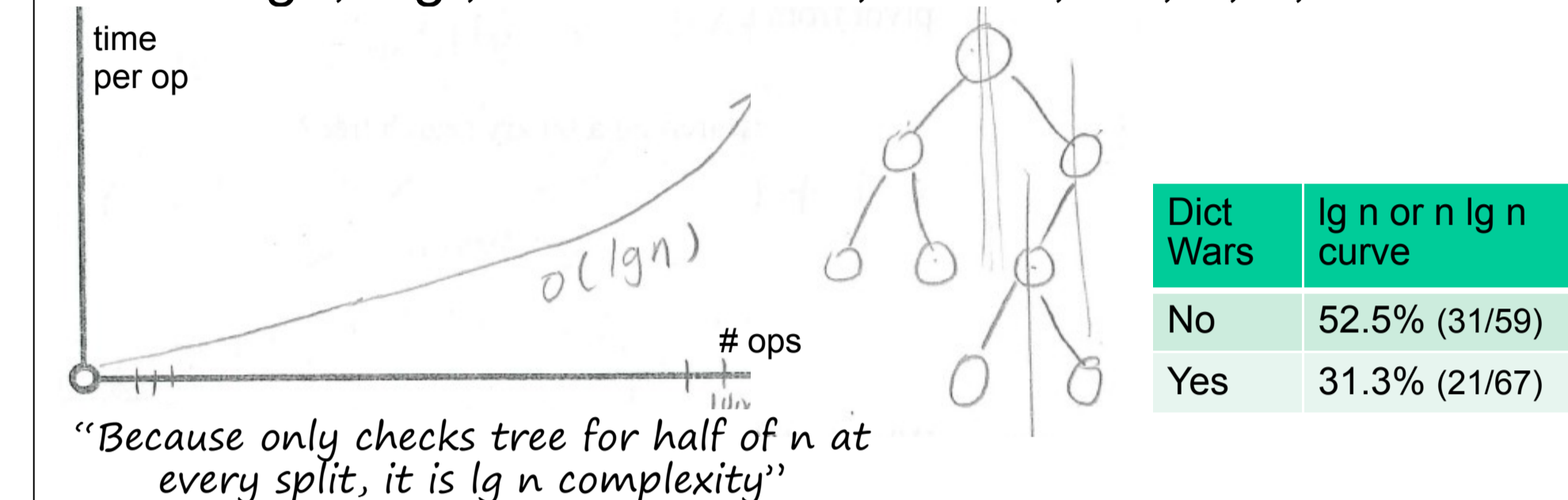
Rank	User	Number of Generators	Total Points	Average Confidence
1	fsy7	10	27.00080	96.6%
2	ky7	10	27.018345	96.5%
3	g1f7	10	26.921222	95.8%
4	epw7	10	25.596236	91.4%
5	l2w7	10	25.002207	89.3%
6	fsy7	10	24.760418	88.4%
7	v8p7	10	24.589655	87.8%
8	g1f7	10	24.589993	87.7%
9	n5u7	7	24.506465	87.5%
10	ksd5	10	24.258346	86.6%
11	x4u7	10	24.095900	86.1%
12	m7s7	8	24.061190	85.9%
13	v9q7	9	24.017995	85.8%
14	z8p7	10	23.928820	85.5%
15	m7B	9	23.899875	85.4%
16	jsz5	10	23.881523	85.3%
17	g3f7	10	23.336447	83.3%
18	l5o7	10	23.286914	83.2%
19	l8r7	4	23.282008	83.2%
20	y0f7	5	23.190179	82.8%
21	n7f7	8	23.170915	82.8%
22	fsq7	10	22.983612	82.1%
23	ky7	5	22.933901	81.9%
24	z2v7	10	22.756143	81.3%
25	m6v7	10	22.708707	81.1%
26	v5w7	10	22.705322	81.1%
27	l1f7	10	22.699973	81.1%
28	u2y7	10	22.665277	80.9%
29	z3u7	9	22.588308	80.7%
30	q7f7	7	22.467773	80.2%



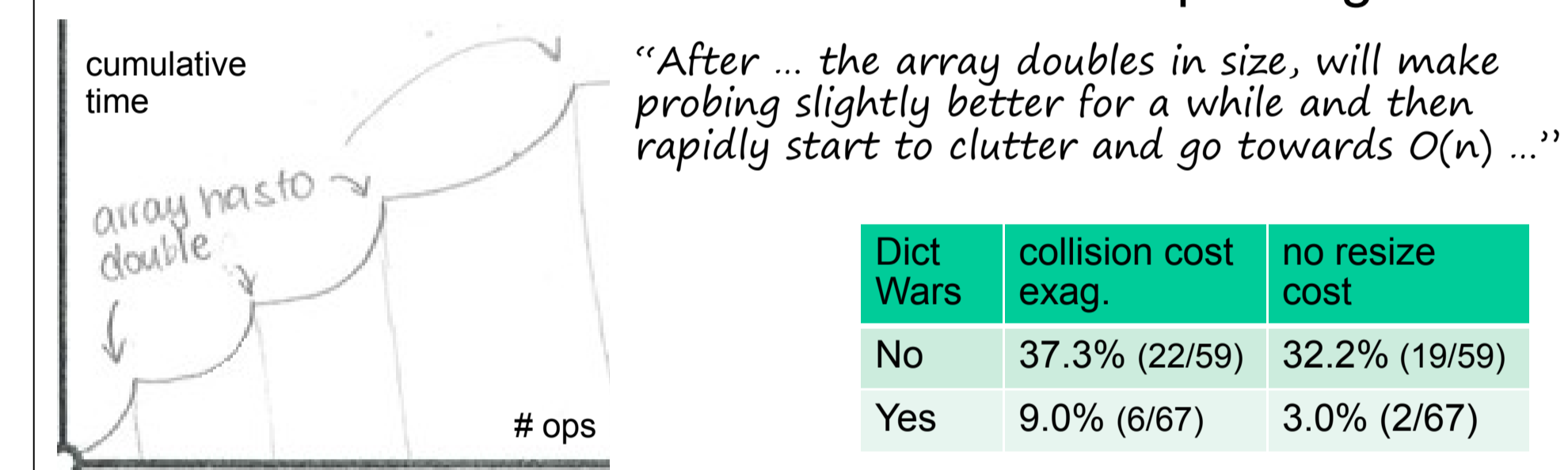
Misconceptions Addressed

Methodology: Two terms of data structures courses with the same question on final exams: graphing data structure performance for inputs. 1st term used DictWars; 2nd did not. We identified patterns in a sample from each term and coded all exams for presence/absence of key patterns.

Attention to context: Ignoring details, relying on general knowledge, e.g., insert 100000, 99999, ..., 3, 2, 1 in BST:



Practical knowledge of performance: Overlooking key or exaggerating minor performance issues, e.g., collision/resize costs in hash table w/linear probing:



Conclusions

Dictionary Wars is a reusable project about asymptotic and practical behaviour of dictionary data structures in which each student receives a custom puzzle to solve. Students enjoy the puzzle nature, quick feedback, and mild competition. Initial results suggest the project improves students’ understanding of data structures’ practical performance and behavior in context. **Potential adopters should e-mail the contact authors.**

Current requirements:

- **Assignment management server:** C++0x/C++11 compiler, bash for setup/grading scripts, distribution of student exes. (e.g., by web or file system access)
- **Student submission analysis server:** C++0x/C++11 compiler, bash, unzip, recent python, account safe for running student code (e.g., a slightly mod’d Amazon Web Services Linux micro instance)
- **Web-based leaderboard server:** sftp access via ssh-keys, basic web serving capabilities
- **Students:** run exes. produced by assn. mgmt. server

Data structures: unsorted linked list (optionally “move to front”, various semantics for duplicates), sorted vector, binary search tree (optionally w/tombstones), AVL tree (optionally w/tombstones), hash table (non-resizing chaining or open addressing w/linear or quadratic probing), splay tree, binary min-heap (not a dictionary; ignores find/remove parameter).

Assignment Writeup

Look in ~course/assigns/dictwars. There’s an executable with your name on it. (Well, login ID, actually.) Looks like a mystery to me. **Solve it!**

“Full” Assignment Writeup

We expand on that a bit to tell students:

- That each executable is different.
- How to copy, run, and get help from the executable.
- What the input language for the executable is.
- A simple sample input generator.
- Required format for their auto-graded permutation.
- Looser required format for their report.
- A full rubric for grading their submission.
- Tips on generating graphs.
- How to access the web server.

Plus 1-2 milestone submissions in which students find the “easy” dictionaries, practice submission and writing reports, and receive interim feedback.