Using Alternative Grading in a Non-Major Algorithms Course

Robbie Weber

University of Washington, Seattle

rtweber2@uw.edu weberrobbie.com

Background: Alternative Grading

- Don’t just “add up” points.
  - Directly ask to what degree learning objectives were met.
  - With points: 3 good-faith, but wrong attempts get 67% each and 2 full-credit submissions and 1 blank submission get the same grade.
- Many versions in the literature, often overlapping
  - Standards-based, specifications-based, mastery-based, etc.
- Common key tenets
  - Simplify grading per assignment (fewer possible outcomes)
  - Evaluate learning throughout term.

Summary and Context

- We ran a senior-level algorithms course for non-majors with a standards-based grading system.
- 70-person course
- Traditional topic list, but math background was highly varied.
- Students in math, informatics, various engineering; both undergraduates and master's students.

Grading Scheme

- 8 homeworks, no exams or other graded material.
  - Each homework expected 1 “mechanical problem” and 3 “long-form” problems from students.
- Up to 2 “old” problems can be (re-)submitted every week.
- Additionally, (auto-graded) programming questions can be resubmitted at any time.

Grading Scheme

- To keep grading load under control, simplify grading.

Excellent	Main idea and edge cases are all correct. Would have gotten full credit (or extremely close) with points-based grading.
Satisfactory	Main idea is correct, but some edge cases or follow-up questions are wrong or missing. Would have gotten about 80-90% on points-based grading.
Not Satisfactory	Some important error is made, but substantial progress toward a solution. Would have gotten above 50% on points-based grading.
Ungraded	Directions not followed (e.g., used a library that isn't permitted) or otherwise shows no substantial progress.

Grading Scheme

Grade	Mechanical Problems	Long-Form Problems
4.0	6E and 2S	22E and 1S
3.5	6E and 1S	18E and 4S
3.0	5E and 1S	14E and 3S
2.5	4E and 1S	8E and 8S
2.0	2E and 2S	6E and 7S
0.7	none	10S, with at least one on five different HWs.

Requirements announced before start of quarter.
Between major grade-breaks, “interpolate” with a formula made at the end of the quarter.

Goals

- Learning that happens later in the quarter should be rewarded.
  - Without a final exam.
- But without overwhelming TAs with grading.
- Students should see benefits of system.
- Allow easier customization of learning objectives for students.

Goals

- **Learning that happens later in the quarter should be rewarded.**
  - **Without a final exam.**
- But without overwhelming TAs with grading.
- Students should see benefits of system.
- Allow easier customization of learning objectives for students.

Most students needed resubmissions

Initial grading of DP long-forms

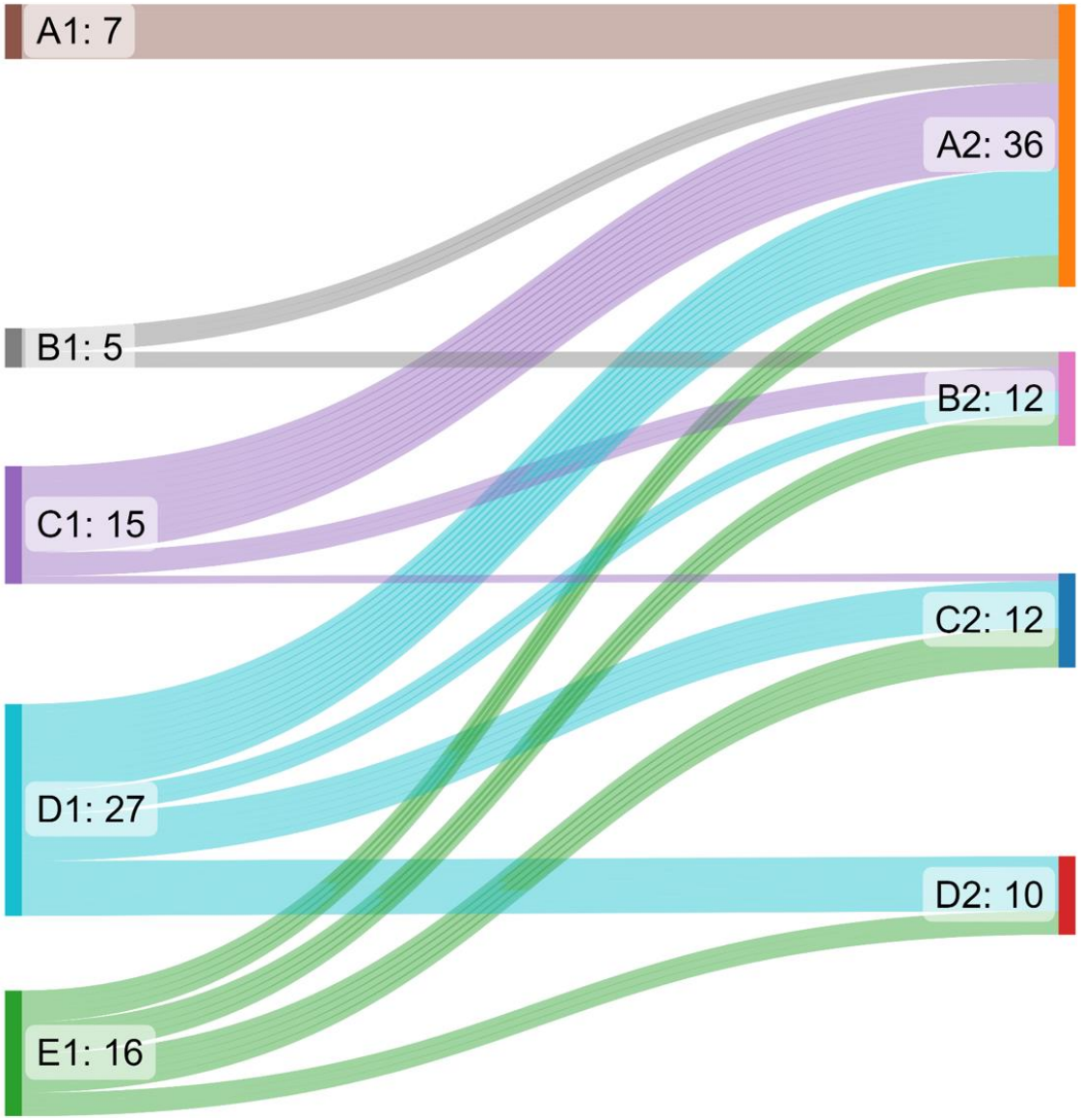
3E scores (perfect)

2E, 1S score

2S scores

1 S score

0S scores (“no progress”)



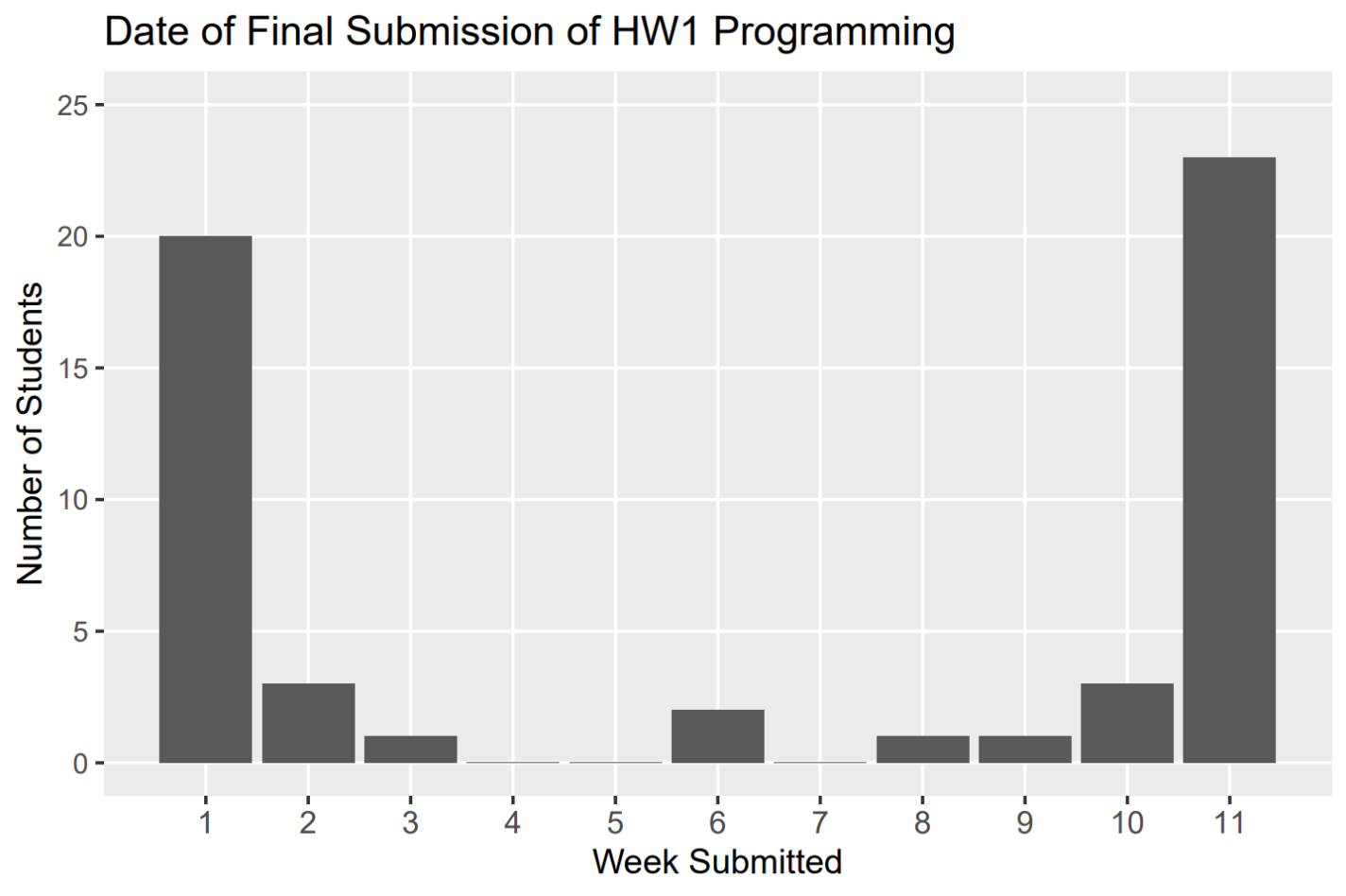
End-of-term grades of DP long-forms

80% of students who finished with all E’s needed resubmission.

No students finished with no S scores.

Figure 2

Learning Throughout the Quarter



Students also learned via programming questions throughout the quarter.

Though procrastination is certainly also a factor here.

Figure 1

Goals

- Learning that happens later in the quarter should be rewarded.
  - Without a final exam.
- **But without overwhelming TAs with grading.**
- Students should see benefits of system.
- Allow easier customization of learning objectives for students.

TA Feedback

- Instead of 4 problems per student per week, we had (on average) about 5.2 problems submitted per student per week.
  - some autograded.
- But grading is simplified (just E/S/N/U, not points-out-of-20)
- TAs did not feel overworked, were comfortable increasing from 1 resubmission to 2 mid-quarter.
- Issues other than grading arose
  - Office hours are much harder!

Goals

- Learning that happens later in the quarter should be rewarded.
  - Without a final exam.
- But without overwhelming TAs with grading.
- **Students should see benefits of system.**
- Allow easier customization of learning objectives for students.

Student Feedback

- Surprisingly uncontroversial
  - Other UW CSE courses have used similar systems.
  - Mid-quarter feedback session had consensus like the system.
  - Few comments on end of quarter evaluations; most positive.
- Keeping up “student morale” needs different strategies.
  - Getting an N is getting a 0, not, say 50% of points.
- Students need more “homework style resources” (no solutions)
- Course-grade assignment process needs to be very clear.

Goals

- Learning that happens later in the quarter should be rewarded.
  - Without a final exam.
- But without overwhelming TAs with grading.
- Students should see benefits of system.
- **Allow easier customization of learning objectives for students.**

Customize Learning

- Student group has a lot of populations:
  - Math majors with extra background.
  - Undergrads about to apply to tech industry.
  - Master's students looking for skills applicable-to-research
- In a "standard" course, hard to treat each group well.
- With simplified grading, can add an "extra" problem for students to choose among more easily.
- Students did customize!
  - Some did every programming question; others did the minimum.
  - Some chose optional "real-world impacts" questions

Fit for the course

- Algorithms problems naturally fit simplified grading well.
- Non-majors is a particularly good fit
  - Optional problems are easier to insert.

Summary

- Learning that happens later in the quarter should be rewarded.
  - Without a final exam.
- But without overwhelming TAs with grading.
- Students should see benefits of system.
- Allow easier customization of learning objectives for students.

We met all of those goals!

But there are quirks and unique difficulties to address with the alternative system.