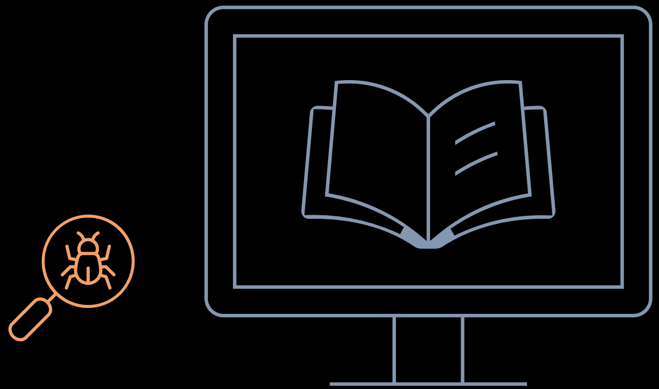
Eliciting Course Feedback through a Bug Bounty Program

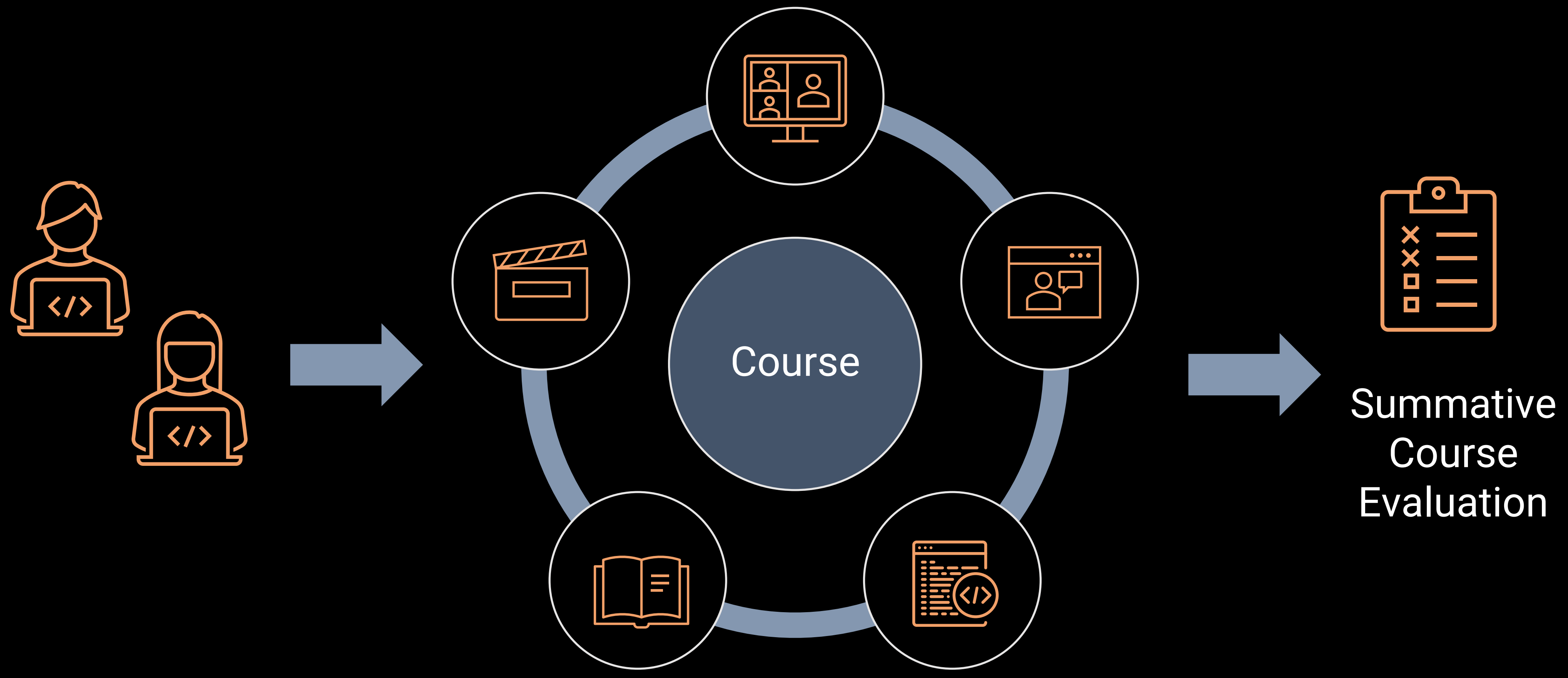


Amanpreet Kapoor [1, 2], Andrew Penton [2], Hamish Pierpont [2]

1: Department of Engineering Education & 2: Computer and Information Science and Engineering Department

University of Florida

Status Quo



Course

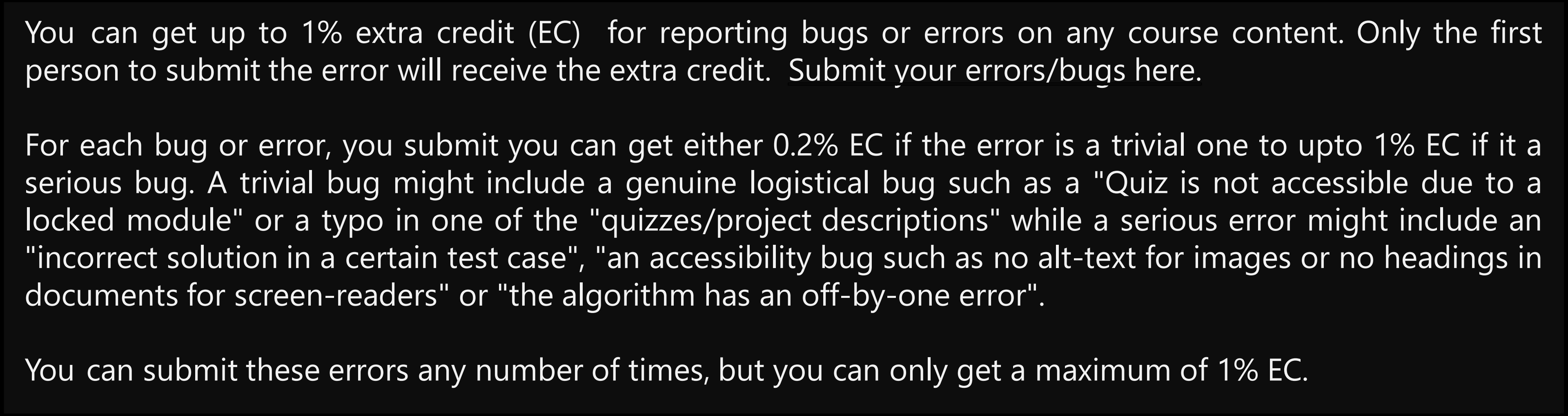
Summative Course Evaluation

Problems with this approach

- Students impacted by:
  - Recency bias and retention in memory
  - Authority bias (afraid to state that something is not right)

- Instructors impacted by:
  - Information overload in large classes
  - Systematic extraction on what to fix

Bug Bounty Program

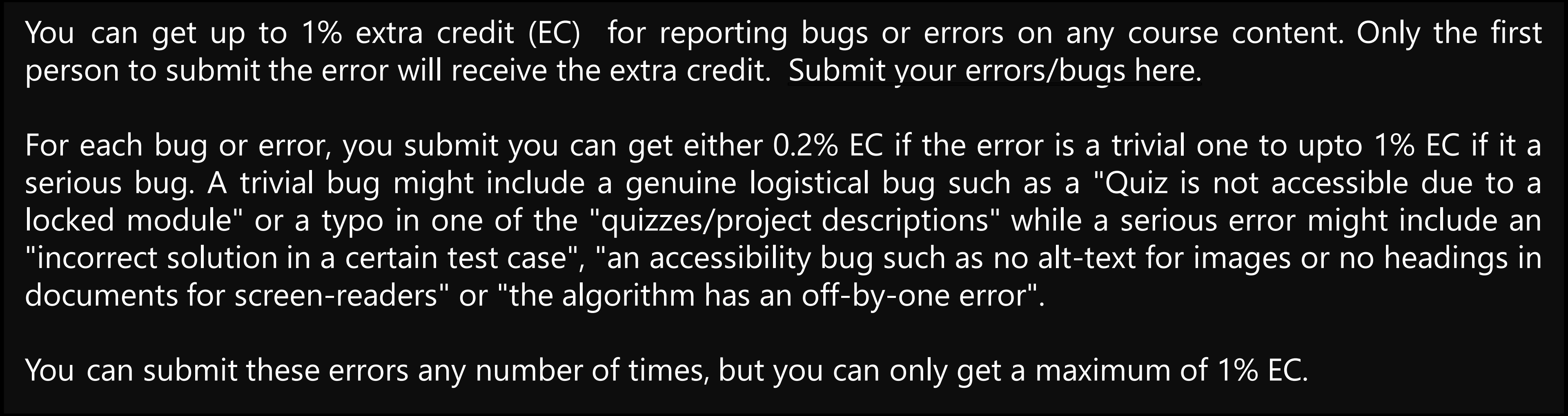


You can get up to 1% extra credit (EC) for reporting bugs or errors on any course content. Only the first person to submit the error will receive the extra credit. Submit your errors/bugs here.

For each bug or error, you submit you can get either 0.2% EC if the error is a trivial one to upto 1% EC if it a serious bug. A trivial bug might include a genuine logistical bug such as a "Quiz is not accessible due to a locked module" or a typo in one of the "quizzes/project descriptions" while a serious error might include an "incorrect solution in a certain test case", "an accessibility bug such as no alt-text for images or no headings in documents for screen-readers" or "the algorithm has an off-by-one error".

You can submit these errors any number of times, but you can only get a maximum of 1% EC.

Bug Bounty Program



You can get up to 1% extra credit (EC) for reporting bugs or errors on any course content. Only the first person to submit the error will receive the extra credit. Submit your errors/bugs here.

For each bug or error, you submit you can get either 0.2% EC if the error is a trivial one to upto 1% EC if it a serious bug. A trivial bug might include a genuine logistical bug such as a "Quiz is not accessible due to a locked module" or a typo in one of the "quizzes/project descriptions" while a serious error might include an "incorrect solution in a certain test case", "an accessibility bug such as no alt-text for images or no headings in documents for screen-readers" or "the algorithm has an off-by-one error".

You can submit these errors any number of times, but you can only get a maximum of 1% EC.

Prompts on Forms:

1. Content (example: lecture video link, project 1 handout, etc.)
2. Error/bug description
3. Name (optional, required for EC)
4. Email (optional, required for EC)

Context

- Large Data Structures and Algorithms Course at a public university in the United States
- Data from four consecutive semesters: Summer 2020 to Summer 2021
- Course content under development over Summer and Fall 2020
- Hybrid mode with a mix of online recordings and synchronous lectures and discussions
- Students were offered up to 1% Extra credit for reporting bugs

Participation Statistics

200

23% Students reported bugs (N=898)

197

99% Students reported bugs for extra credit

373*

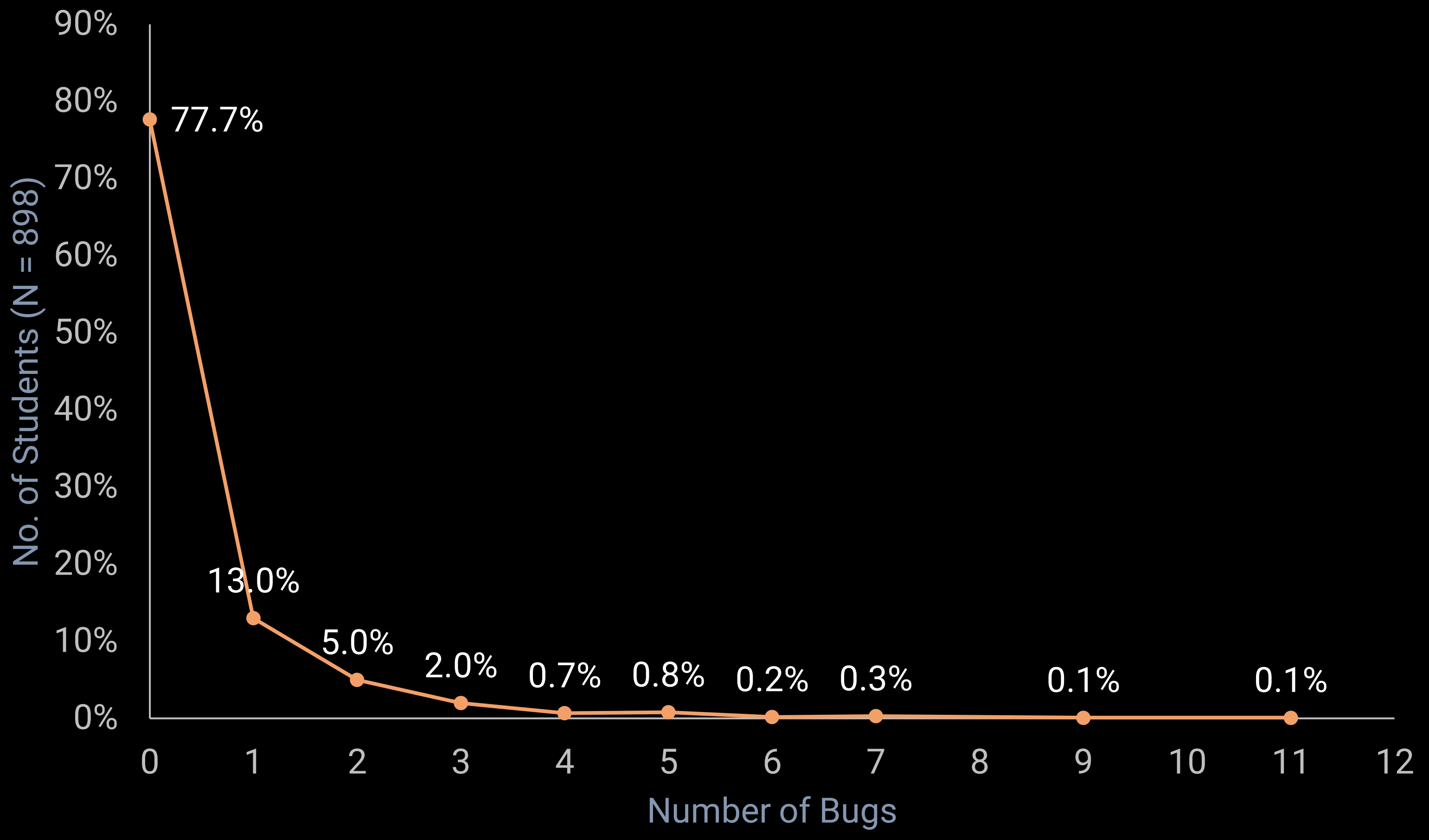
Reported bugs after removing 10 multiple entries of a bug by a student

* Corpus for this paper

307

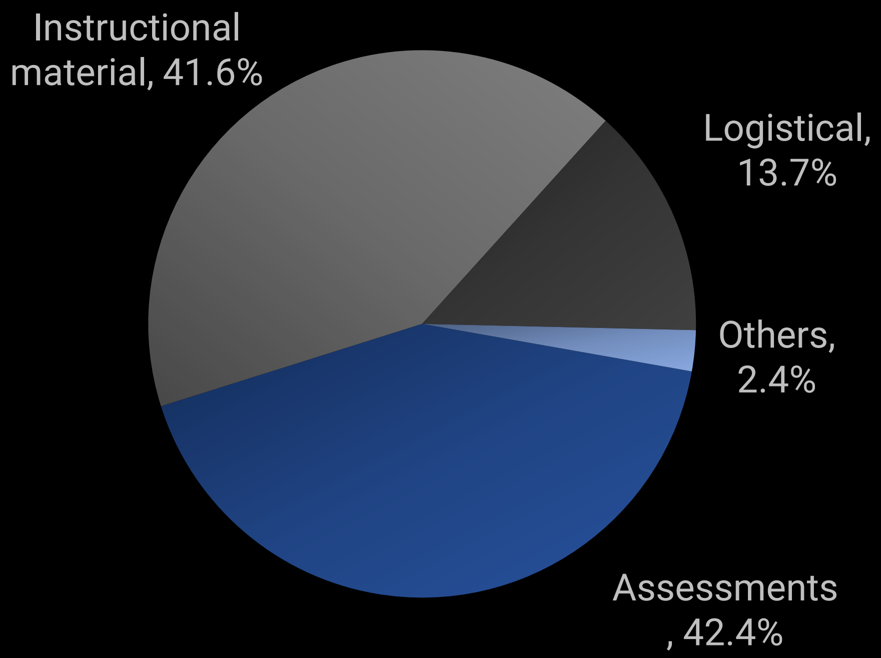
82% Unique bugs (18% redundancy)

Bugs reported per student

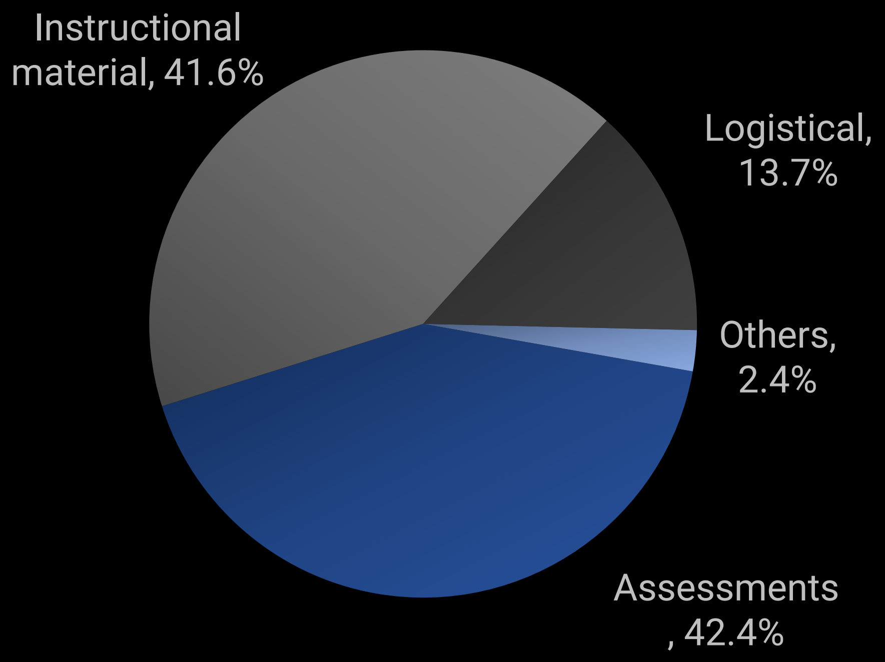


# Bugs	# Students (N = 898)
0	698
1	117
2	45
3	18
4	6
5	7
6	2
7	3
9	1
11	1

Type of content impacted by a bug



Type of content impacted by a bug



Category	Subcategory	Count (N=373)	Category Count	%
Assessments	Coding problems	68	158	42.4%
	Exam	9		
	Project	26		
	Quiz	55		
Instructional material	Polling	3	155	41.6%
	Slides	66		
	Videos	86		
Logistical	Logistical	51	51	13.7%
Others	Others	9	9	2.4%

Types of Bugs

- 58 types of bugs were reported

- Examples include
  - Typos
  - Ambiguous content
  - Accessibility bugs, e.g., missing alt texts
  - Broken hyperlinks
  - Lecture miscommunications
  - Inexhaustive testing of a programming problem
  - Not a bug and other misconceptions
  - Due date issues
  - Platform discrepancies

Types of Bugs: Typo

In quiz 2 it says:

```
int x = 1;
while (x < n)
{
     x * 2;
}
```

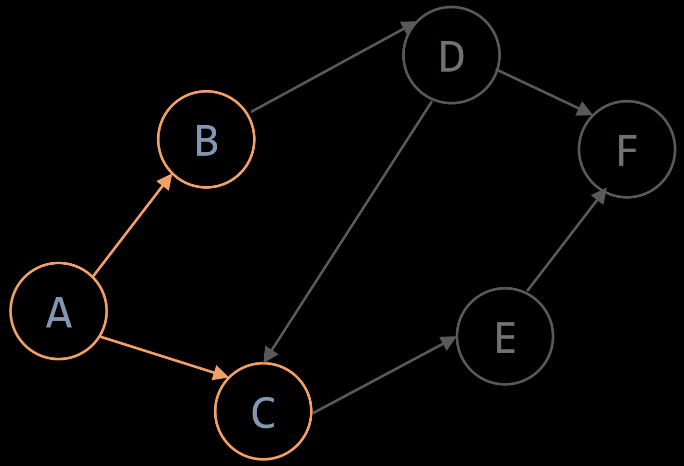
it should say, `x *= 2;`

Types of Bugs: Inexhaustive Testing

Some edge cases are not captured by test cases. For example, my code works for all 8 test cases (9 is fake), but fails on an array of even size where the last element in the array does not follow the min/max heap data structure. For example, arr = [10, 17, 22, 23] is a min/max heap, however arr = [10, 17, 22, 2] is not a min or max heap. It appears test cases on stepik do not cover this.

Types of Bugs: Not a Bug

These two videos contradict each other when talking about DFS, in 5h it says that the DFS is ABDFCE but in 5i the slide says the DFS is ABDCEF

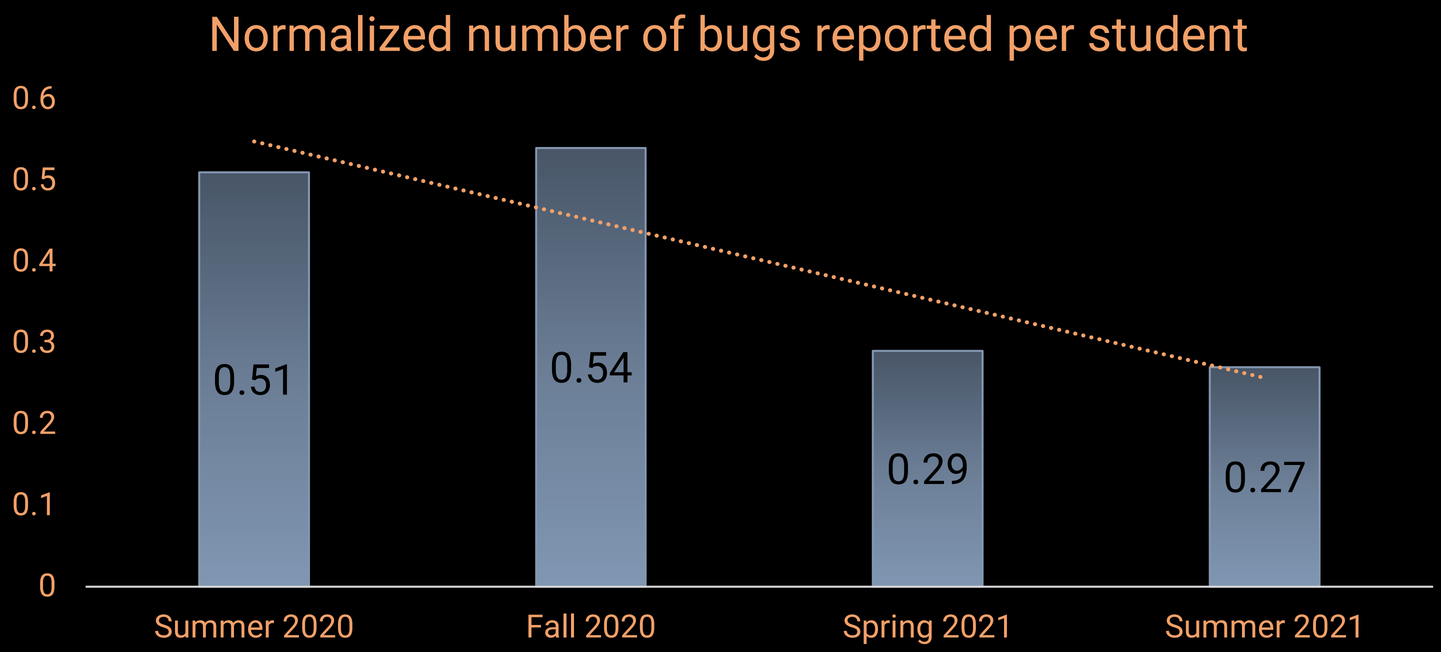


Types of Bugs: Platform discrepancies

This problem's Test case #6 requires you to realize that two large integers might cause an overflow so you should cast the sum to a bigger sized type. In the lecture, Prof. Kapoor asked the class what should we cast the sum to. I answered that we should cast it to a long and he agreed. He proceeded to finish the problem by casting it to a long. Stepik accepted it. On x86_64, long's are 64-bit so this works exactly as discussed in class. However, on some architectures and operating systems like Linux 32-bit (x86) `long int` and `int` allocate the same amount of memory (32 bits). So the sum should be casted to a `long long` to assure that you are using a 64-bit variable in all architectures/OSs.

References for `long int` allocating different memory depending on system architecture:
https://en.wikipedia.org/wiki/64-bit_computing#64-bit_data_models
https://en.cppreference.com/w/cpp/language/types

Efficacy



Semester	Course strength	# bugs	Bugs reported per student
Summer 2020	143	73	0.51
Fall 2020	333	179	0.54
Spring 2021	244	73	0.29
Summer 2021	178	48	0.27

Student Reception

How was your experience in the bug bounty program? Should it be a part of future course offerings?

46%

Positive

"Bug bounty was extremely helpful both for the students and the instructors I believe. Students could clear up any misconceptions and get extra credit in the process, while the instructor is notified of their errors"

Student Reception

How was your experience in the bug bounty program? Should it be a part of future course offerings?

51%

Neutral

“I did not encounter any bugs to report, so I am neutral”

Student Reception

How was your experience in the bug bounty program? Should it be a part of future course offerings?

3%

Negative

“I do not think the bug bounty program should be a part of the course because it forces students to try to find errors and distracts from other things”

Recommended Practices

- Show students bugs reported by others to avoid redundancy
- Give them extra credit for participation
- Use appropriate links to report bugs on Course Homepage

Questions

✉ kapooramanpreet@ufl.edu