

# Conceptual Mutation Testing for Student Programming Misconceptions



Siddhartha Prasad  
Ben Greenman  
Tim Nelson  
Shriram Krishnamurthi



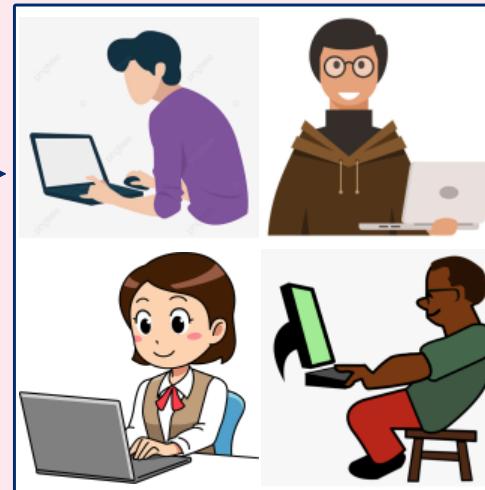
BROWN



## Understanding CS Problems

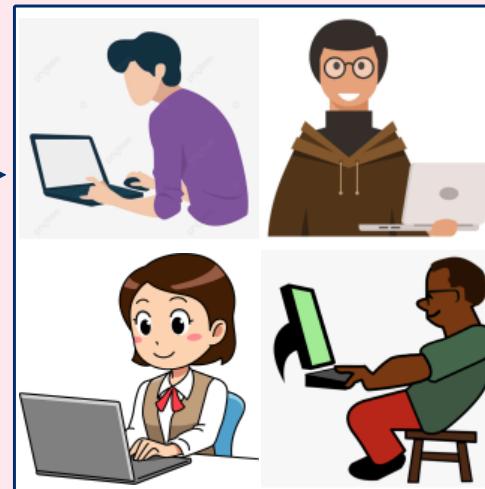
## Understanding CS Problems

Q. median



## Understanding CS Problems

Q. median



## Understanding CS Problems

Q. median

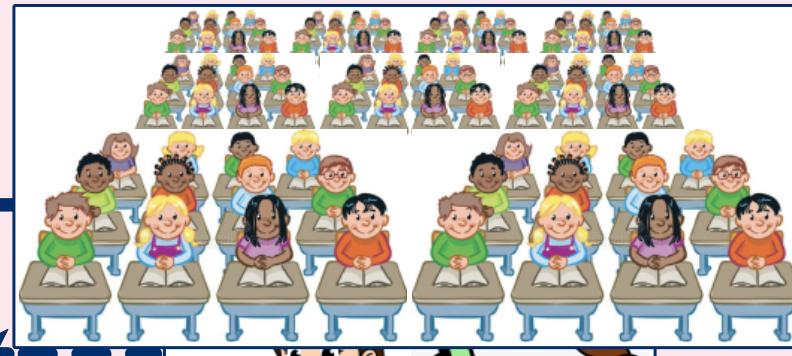


feedback?



## Understanding CS Problems

Q. median

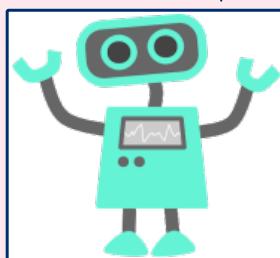
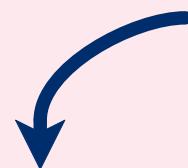
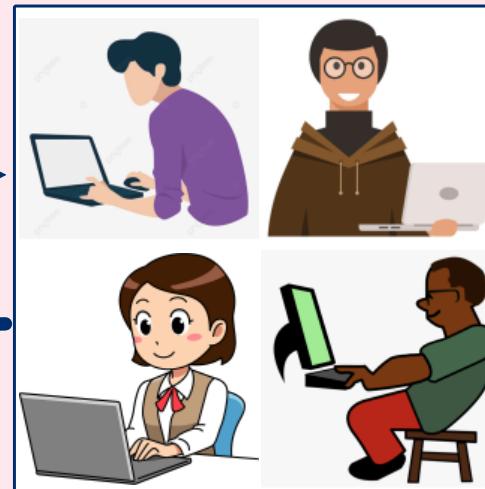


feedback?



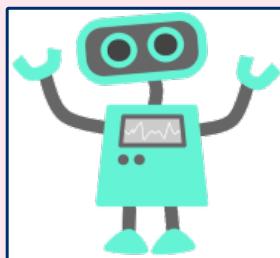
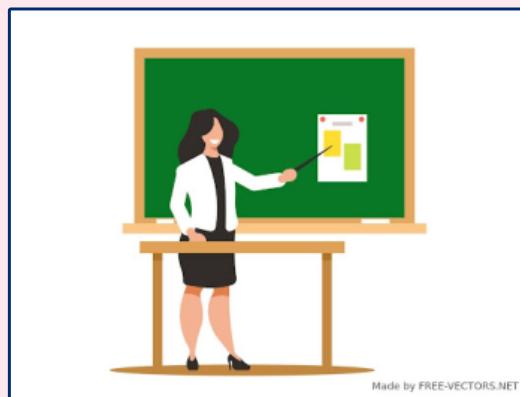
## Understanding CS Problems

Q. median

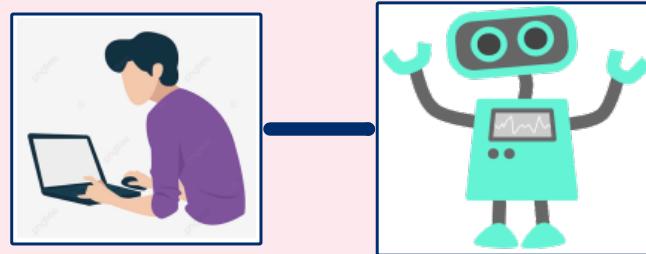


## Understanding CS Problems

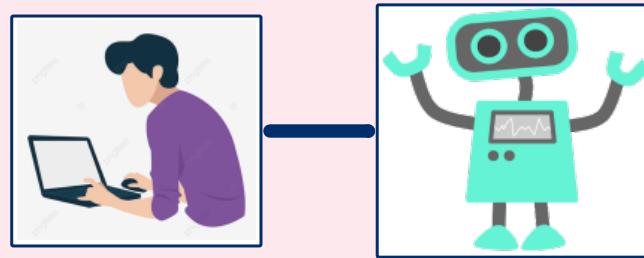
Q. median



How to communicate?



How to communicate?

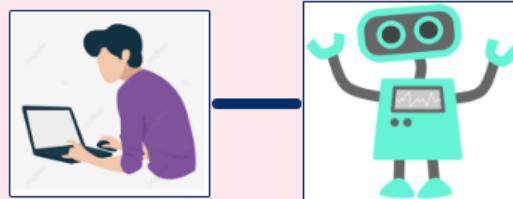


Test Cases

Q. median



**Q. median**



`median [1] is 1`

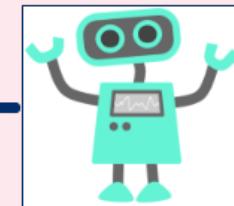
`median [1, 2, 3] is 3`

`median [3, 3, 3] is 3`

Q. median



Made by FREE-VECTORS.NET



median [1] is 1

median [1, 2, 3] is 3

median [3, 3, 3] is 3

What's **wrong** with these tests?



View File median-tests.arr + Begin

```
1 v include my-gdrive("median-code.arr")
2 # DO NOT CHANGE ANYTHING ABOVE THIS LINE
3
4 v check:
5 v   median([list: 1]) is 1
6 v   median([list: 1, 2, 3]) is 3
7 v   median([list: 3, 3, 3, 3]) is 3
8
9
10 end
```

The screenshot shows a programming interface with a code editor and a test results window.

**Code Editor:**

```
1 include my-gdrive("median-code.arr")
2 # DO NOT CHANGE ANYTHING ABOVE THIS LINE
3
4 check:
5 median([list: 1]) is 1
6 median([list: 1, 2, 3]) is 3
7 median([list: 3, 3, 3, 3]) is 3
8
9
10 end
```

**Test Results Window:**

median-tests.arr

**INCORRECT** CONSEQUENTLY, THOROUGHNESS IS UNKNOWN

These tests do not match the behavior described by the assignment:  
definitions://5:2-5:30

```
6 | median([list: 1, 2, 3]) is 3
```



View File median-tests.arr + Begin

```
1 ▾ include my-gdrive("median-code.arr")
2 # DO NOT CHANGE ANYTHING ABOVE THIS LINE
3
4 ▾ check:
5 ▾   median([list: 1]) is 1
6 ▾   median([list: 1, 2, 3]) is 2
7 ▾   median([list: 3, 3, 3, 3]) is 3
8
9   #Shows that Median is not Mean
10 ▾   median([list: 1, 1, 3]) is 1
11
12   # Shows that Median is not Mode
13 ▾   median([list: 1, 1, 3, 4, 4]) is 3
14 end
```

The screenshot shows a software interface with a toolbar at the top. The toolbar includes a logo, View, File, a file tab labeled "median-tests.arr", a "Begin Implementation" button, a Run dropdown menu, and a Stop button.

The code editor on the left contains the following pseudocode:

```
1 v include my-gdrive("median-code.arr")
2 # DO NOT CHANGE ANYTHING ABOVE THIS LINE
3
4 v check:
5 v   median([list: 1]) is 1
6 v   median([list: 1, 2, 3]) is 2
7 v   median([list: 3, 3, 3, 3]) is 3
8
9   #Shows that Median is not Mean
10 v  median([list: 1, 1, 3]) is 1
11
12   # Shows that Median is not Mode
13 v  median([list: 1, 1, 3, 4, 4]) is 3
14 end
```

To the right of the code editor is a results window titled "median-tests.arr". It displays the word "VALID" in large blue letters. Below it, there are four circular icons, each containing a small caterpillar-like character. Two of these icons are filled with blue, while two are empty grey circles.

The results window also contains the following text:

These tests are valid and consistent with the assignment handout.  
They caught 2 of 4 sample buggy programs. Add more test cases to  
improve this test suite's thoroughness.

The screenshot shows a software interface with a toolbar at the top. The toolbar includes a logo, View, File, a file tab labeled "median-tests.arr", a "Begin Implementation" button, a Run dropdown, and a Stop button. The main area has two panes. The left pane is a code editor with the following Python-like pseudocode:

```
1 include my-gdrive("median-code.arr")
2 # DO NOT CHANGE ANYTHING ABOVE THIS LINE
3
4 check:
5   median([list: 1]) is 1
6   median([list: 1, 2, 3]) is 2
7   median([list: 3, 3, 3, 3]) is 3
8
9   #Shows that Median is not Mean
10  median([list: 1, 1, 3]) is 1
11
12  # Shows that Median is not Mode
13  median([list: 1, 1, 3, 4, 4]) is 3
14 end
```

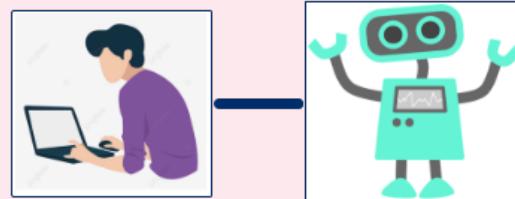
The right pane displays the results of the test suite for "median-tests.arr". It shows a blue header bar with the word "VALID" and four circular icons containing caterpillars. Below this, a message states: "These tests are valid and consistent with the assignment handout. They caught 2 of 4 sample buggy programs. Add more test cases to improve this test suite's thoroughness."

What's **wrong** with these tests?

Q. median



Q. median



Tests must distinguish:  
mean  
median vs. mode  
middle ...

Valid & Thorough

How to check thoroughness?



Buggy solutions  
(mutation testing)



**RQ.** How to design buggies?



**RQ.** How to design buggies?

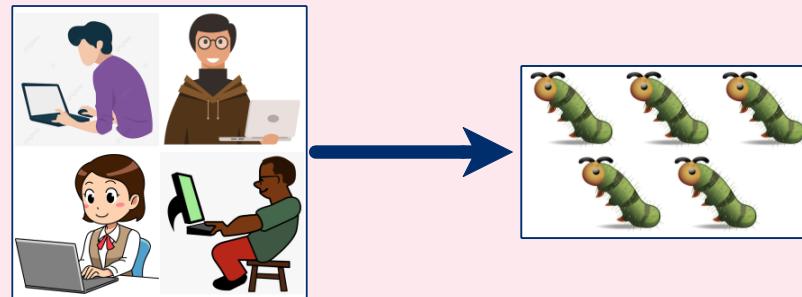


Need to **discover** misconceptions

Prior Work:  
Expert-Driven



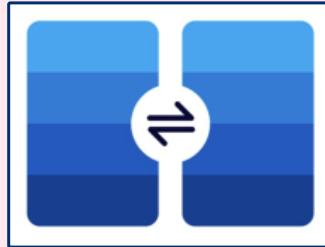
Today, a recipe:  
Buggies from Data



## 1. Design problem

## 1. Design problem

Running example:  
Doc Diff



docdiff ['a'] ['A'] **is** 1

docdiff ['one', 'two'] ['one'] **is** 1/2

docdiff ['hello'] ['world'] **is** 0

2. Collect invalid tests

## 2. Collect invalid tests



median-tests.arr

**INCORRECT** CONSEQUENTLY, THOROUGHNESS IS UNKNOWN

These tests do not match the behavior described by the assignment:

median-tests.arr

**INCORRECT** CONSEQUENTLY, THOROUGHNESS IS UNKNOWN

These tests do not match the behavior described by the assignment.

median-tests.arr

**INCORRECT** CONSEQUENTLY, THOROUGHNESS IS UNKNOWN

These tests do not match the behavior described by the assignment.

median-tests.arr

**INCORRECT** CONSEQUENTLY, THOROUGHNESS IS UNKNOWN

These tests do not match the behavior described by the assignment.

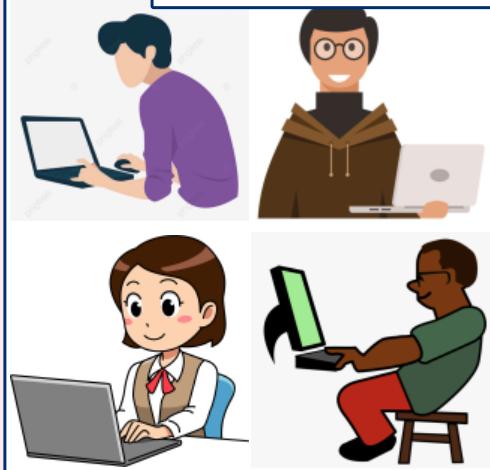
median-tests.arr

**INCORRECT** CONSEQUENTLY, THOROUGHNESS IS UNKNOWN

These tests do not match the behavior described by the assignment.

## 2. Collect invalid tests

Doc Diff ==> 1,500 invalids in ~1 week



median-tests.arr

**INCORRECT** CONSEQUENTLY, THOROUGHNESS IS UNKNOWN

These tests do not match the behavior described by the assignment.

median-tests.arr

**INCORRECT** CONSEQUENTLY, THOROUGHNESS IS UNKNOWN

These tests do not match the behavior described by the assignment.

median-tests.arr

**INCORRECT** CONSEQUENTLY, THOROUGHNESS IS UNKNOWN

These tests do not match the behavior described by the assignment.

median-tests.arr

**INCORRECT** CONSEQUENTLY, THOROUGHNESS IS UNKNOWN

These tests do not match the behavior described by the assignment.

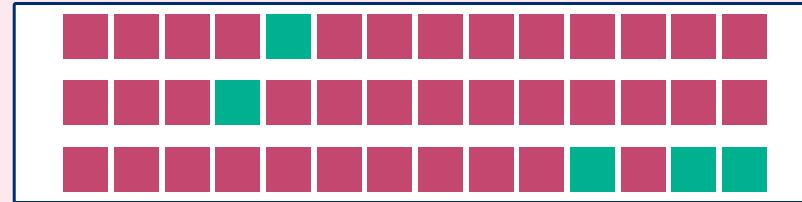
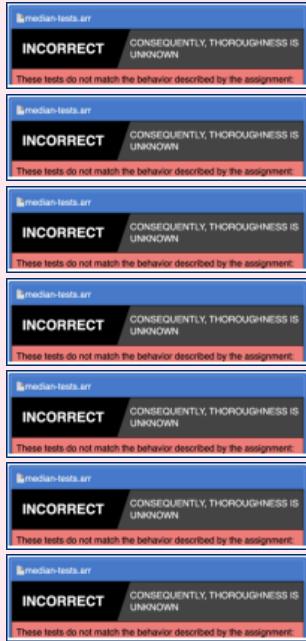
median-tests.arr

**INCORRECT** CONSEQUENTLY, THOROUGHNESS IS UNKNOWN

These tests do not match the behavior described by the assignment.

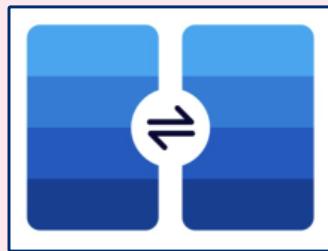
### 3. Cluster tests by feature vector

### 3. Cluster tests by feature vector



Feature vectors  $\leq$  problem characteristics

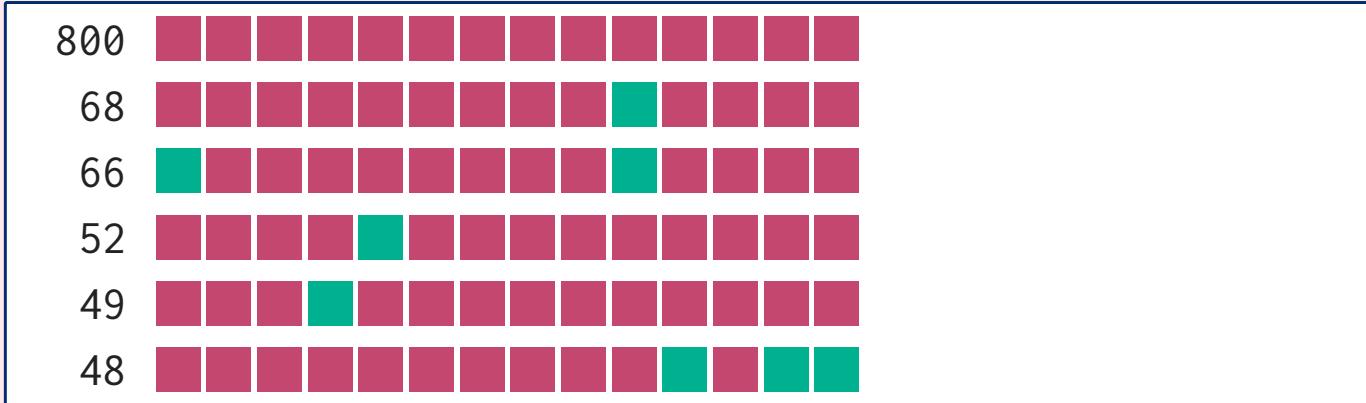
Feature vectors  $\leq$  problem characteristics



- Case-insensitive
- Words may repeat
- Diff may be a fraction
- ... [14 in total]

5. Sort clusters

## 5. Sort clusters



## 5. Sort clusters

800		[typos]
68		diff is always zero
66		case sensitive
52		normalize by wrong mag.
49		normalize by wrong vector
48		[unknown]

5. Sort clusters

6. Make buggies

800



[typos]

68



diff is always zero

66



case sensitive

52



normalize by wrong mag.

49



normalize by wrong vector

48



[unknown]





## 6. Make buggies

- \* Focus on 1-2 
- \* Favor narrow characteristics
- \* Maximize subproblem coverage

## 6. Make buggies

- \* Focus on 1-2 
- \* Favor narrow characteristics
- \* Maximize subproblem coverage

## 6. Make buggies

- \* Focus on 1-2 
- \* Favor narrow characteristics

800



[typos]

68



diff is always zero

66



case sensitive

52



by wrong mag.

49



normalized by wrong vector

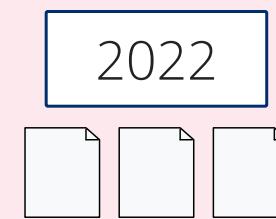
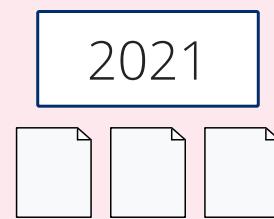
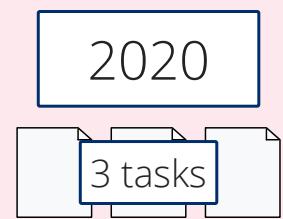
48



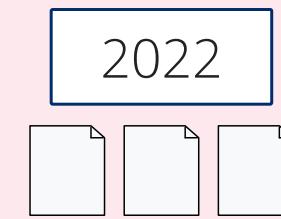
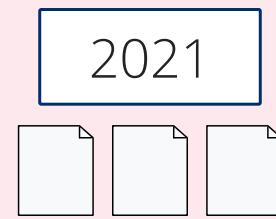
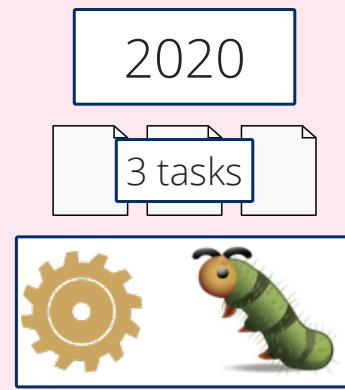
[unknown]

case sensitive is more narrow

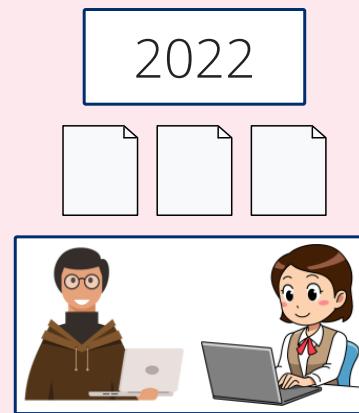
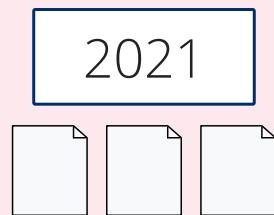
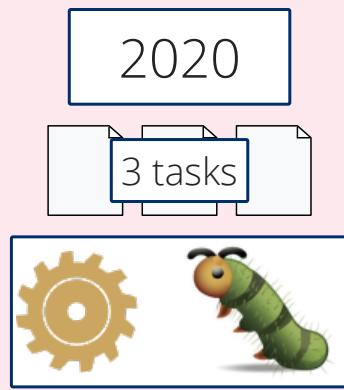
## Evaluation



## Evaluation

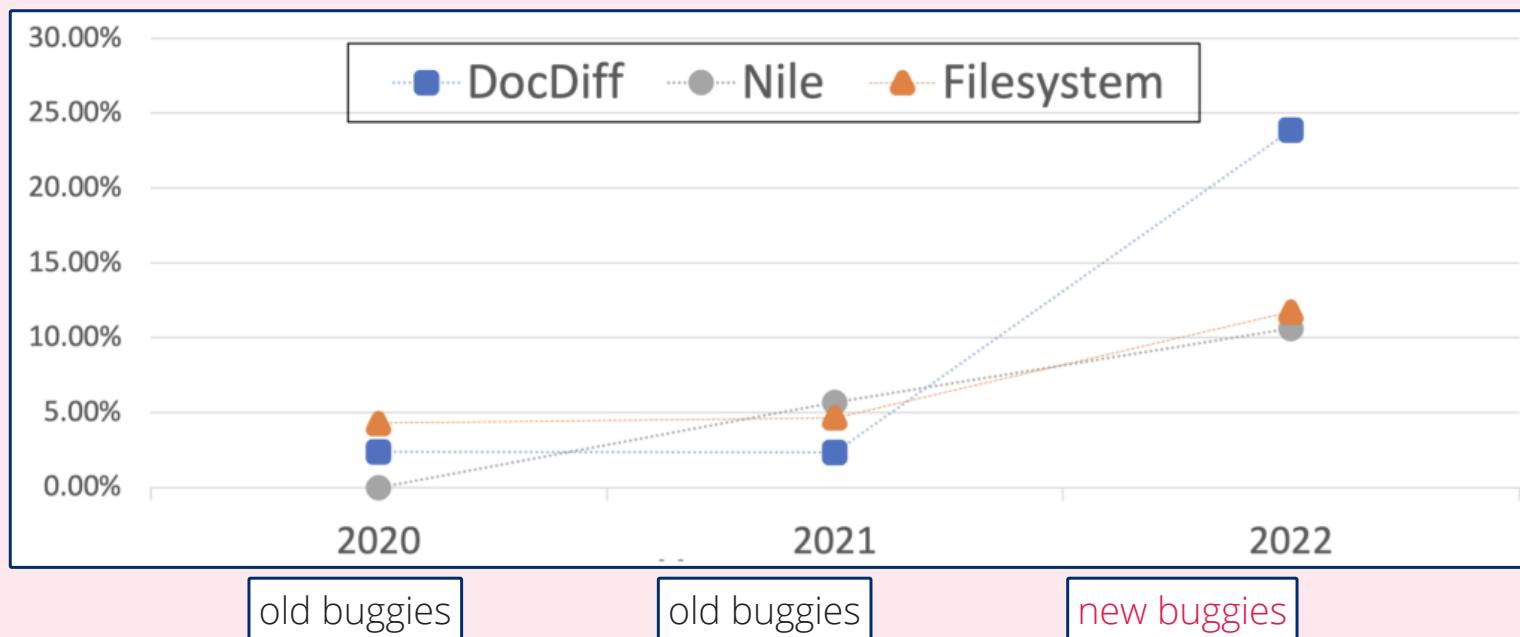


## Evaluation

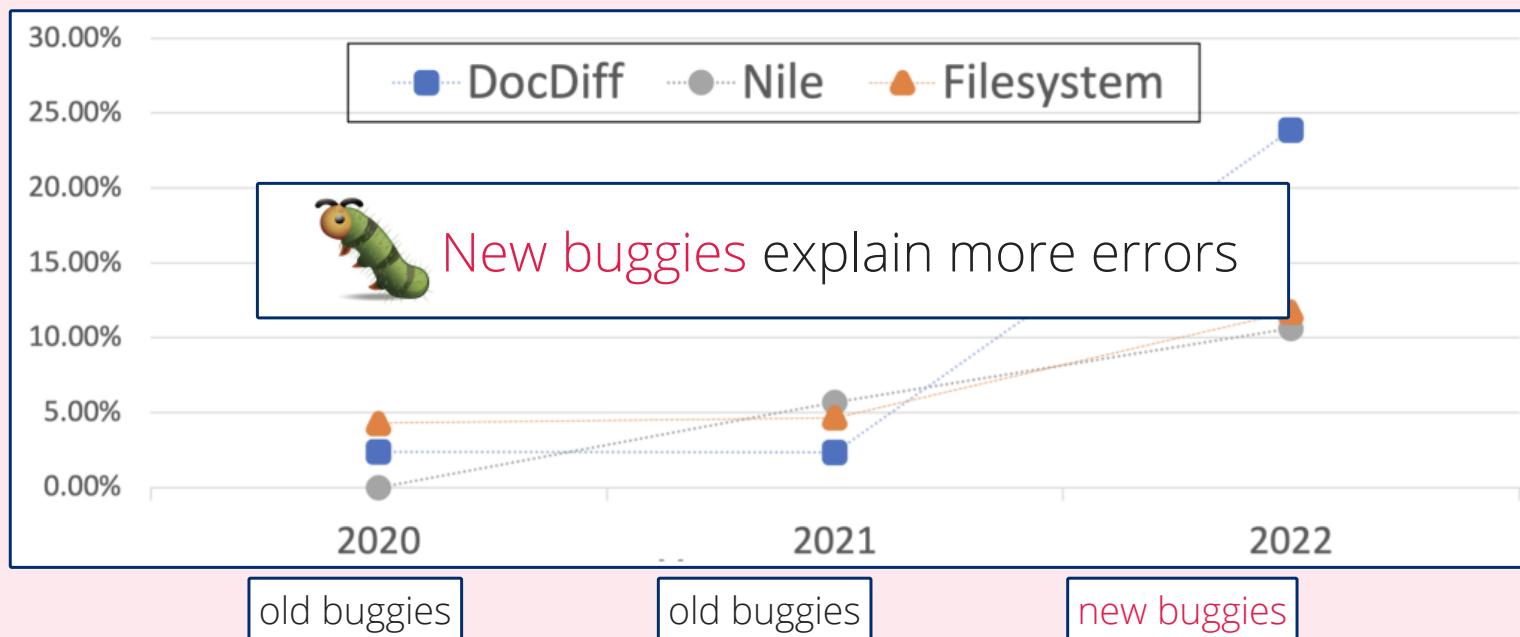


2020, 2021 ==> test  
2022 ==> deploy

% explainable invalid tests  
explainable = 1- or 2-



% explainable invalid tests  
explainable = 1 -  or 2 - 



## High Effect Sizes for 2022



Matchup	Problem	95% CI	p value
2022 vs 2020	DocDiff	<span style="color: red;">[-0.75, -0.57]</span>	1.35E-29
	Nile	<span style="color: red;">[-0.55, -0.26]</span>	9.07E-14
	FileSys	<span style="color: red;">[-0.35, -0.21]</span>	2.35E-10
2022 vs 2021	DocDiff	<span style="color: red;">[-0.70, -0.51]</span>	6.87E-29
	Nile	<span style="color: red;">[-0.27, -0.07]</span>	1.82E-3
	FileSys	<span style="color: red;">[-0.33, -0.19]</span>	2.32E-9
2021 vs 2020	DocDiff	<span style="color: red;">[-0.07, 0.08]</span>	4.60E-1
	Nile	<span style="color: red;">[-0.39, -0.13]</span>	1.15E-17
	FileSys	<span style="color: red;">[-0.06, 0.03]</span>	2.52E-1

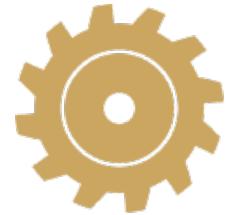
Weeks of  
Data > Years of  
Tuning



Promising approach for new problems



Recipe to uncover misconceptions  
semi-automatic

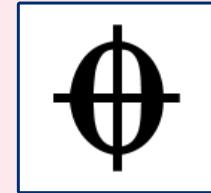


Recipe to uncover misconceptions  
semi-automatic



Data ==> better teaching

What's next? Hinting





What's next? Hinting

docdiff-tests.arr

**INCORRECT**

CONSEQUENTLY, THOROUGHNESS  
UNKNOWN



These tests do not match the behavior described by the assignment.

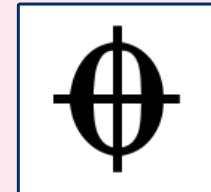
definitions://12:2-12:45

13 overlap([list: "a", "b"], [list: "b"]) **is** 0

**The assignment says:**

Overlap must be proportional to the dot product of two vectors.





Deep Goal:  
Rigorous methods for CS Ed research

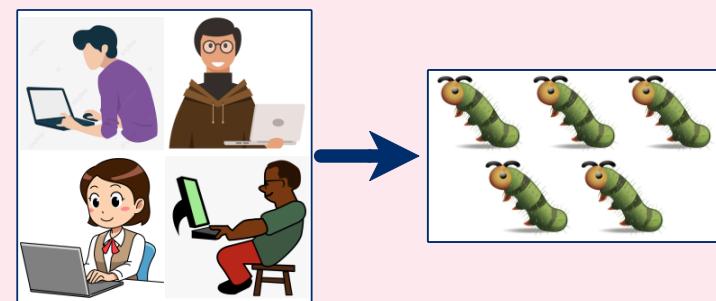




Let's talk!



1. design problem
2. identify characteristics
3. collect invalid tests
4. cluster by feature vector
5. analyze top clusters
6. select buggies



## Future

Data collection is a bottleneck  
~1 semester ramp-up

+70% typos! How to reduce?  
D4 / Data Druid

■ **Table 8** Our 2022 chaffs gave 1-m/2-m outcomes significantly more often than prior chaffs. The 2021 vs. 2020 results are similar except for Nile, which used D4 in 2021.

Matchup	Assignment	p value	Z score	Effect Size [95% CI] (Cohen's D)
2022 vs 2020	DocDiff	1.35E-29	-11.24	0.66 [-0.75, -0.57]
	Nile	9.07E-14	-7.36	-0.41 [-0.55, -0.26]
	Filesystem	2.35E-10	-6.22	-0.28 [-0.35, -0.21]
2022 vs 2021	DocDiff	6.87E-29	-11.09	-0.61 [-0.70, -0.51]
	Nile	1.82E-03	-2.91	-0.17 [-0.27, -0.07]
	Filesystem	2.32E-09	-5.86	-0.26 [-0.33, -0.19]
2021 vs 2020	DocDiff	4.60E-01	0.1	0 [-0.07, 0.08]
	Nile	1.15E-17	-8.48	-0.26 [-0.39, -0.13]
	Filesystem	2.52E-01	-0.67	-0.02 [-0.06, 0.03]



