Learning@Scale 2016

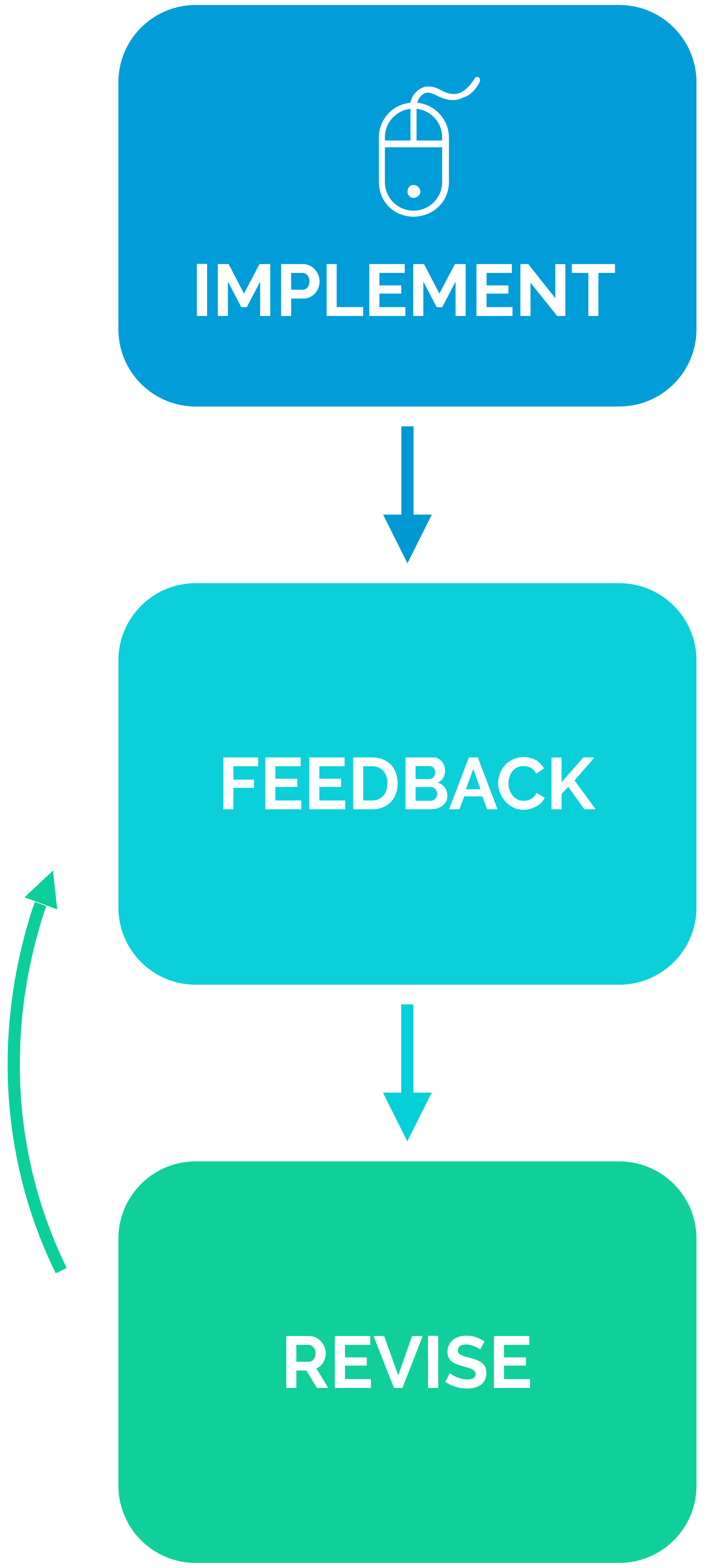
Fuzz Testing Projects in Massive Courses

Sumukh Sridhara, Brian Hou,
Jeffrey Lu and John DeNero

UC Berkeley

April 26th 2016
Edinburgh, UK

Programming Projects in MOOCs


IMPLEMENT

FEEDBACK

REVISE

Programming Projects

1. Primarily Instructional
2. Instructor Solution Exists
3. Automated Feedback

Feedback Goals

1. Help students arrive at a correct answer

2. Students can help themselves

3. Every missed bug is a missed learning opportunity

Targeted

1

Isolates One Issue

Guide Student Attention

Many Targeted Tests

Comprehensive

100%

Tests every case

Hard To Engineer

Hard To Compute

Fuzz Testing

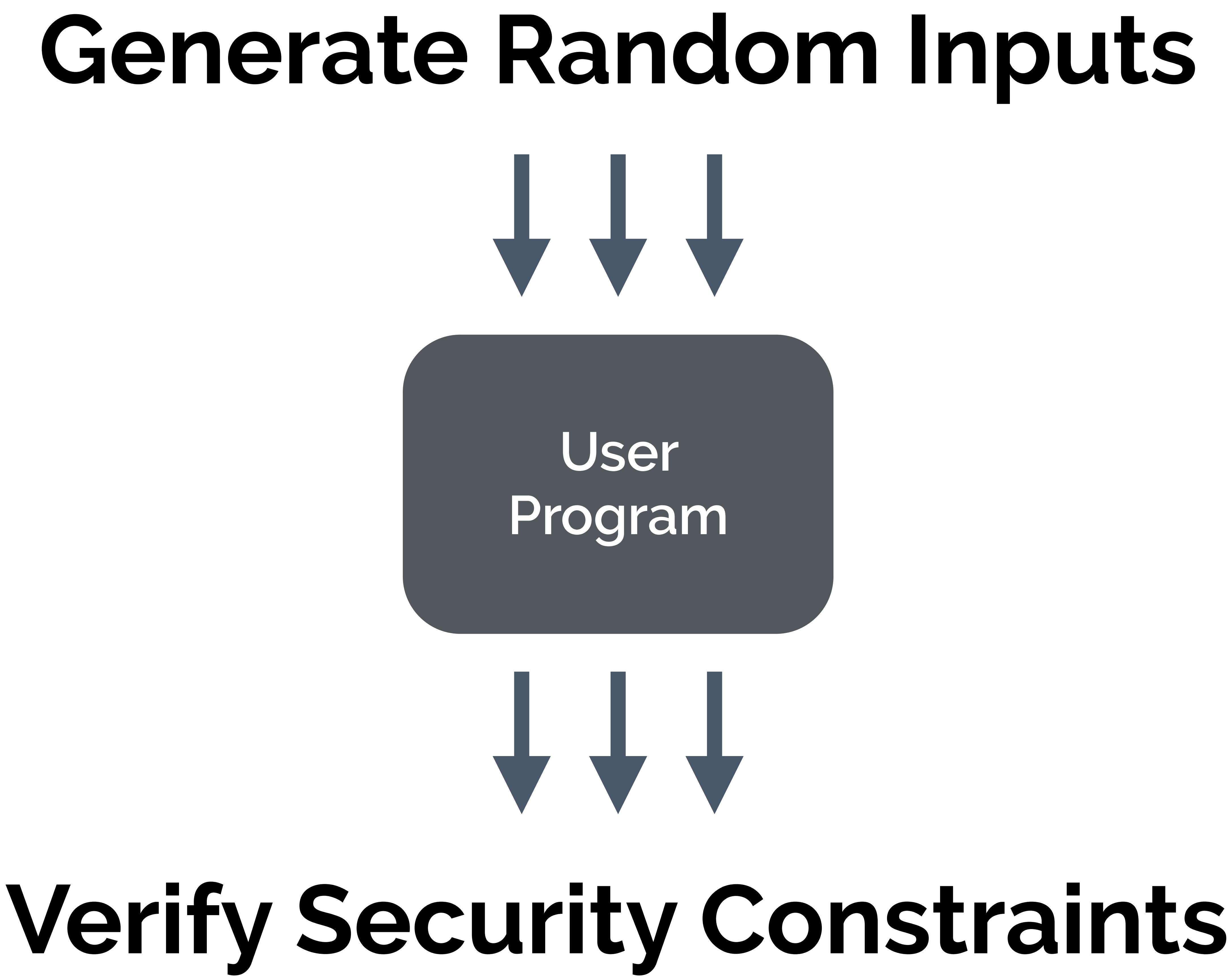
Fuzz Testing

Testing the behavior of the program on many random inputs

Complementary to Manual Testing

Historically used for security

Fuzz Testing (Security)

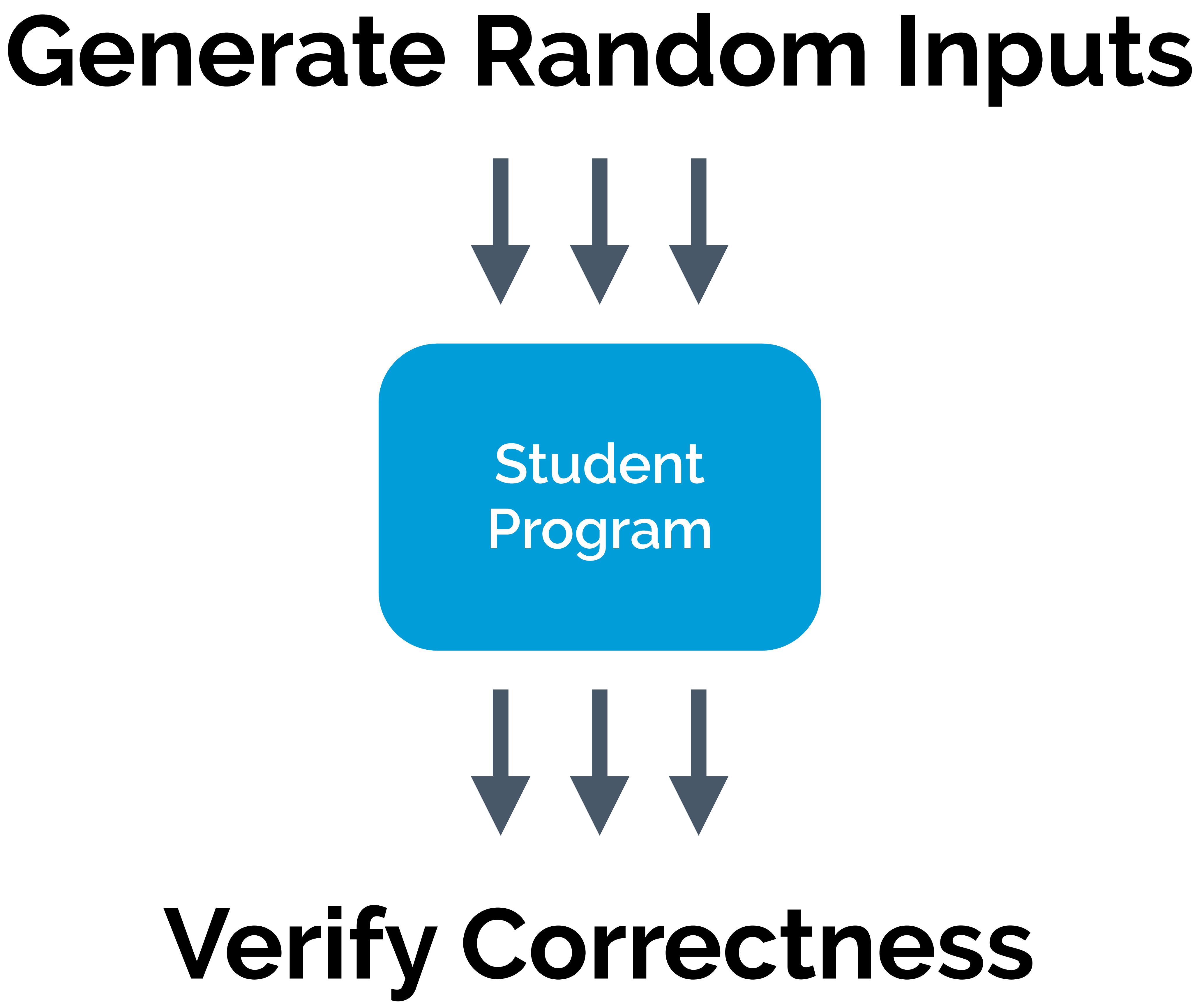


Generate Random Inputs

User Program

Verify Security Constraints

Fuzz Testing (Programming Projects)

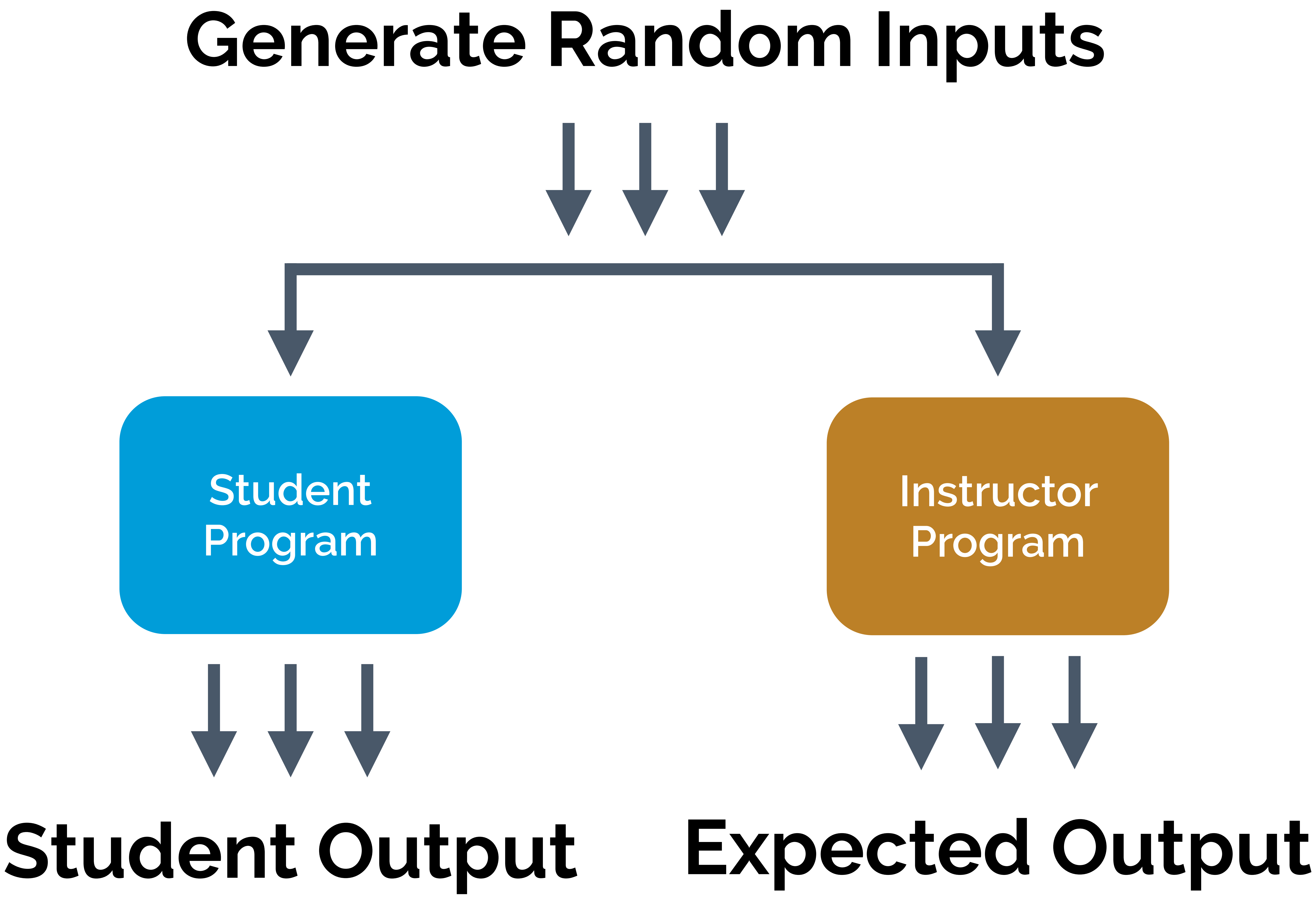


Generate Random Inputs

Student Program

Verify Correctness

Fuzz Testing (Programming Projects)



Generate Random Inputs

Student Program

Instructor Program

Student Output

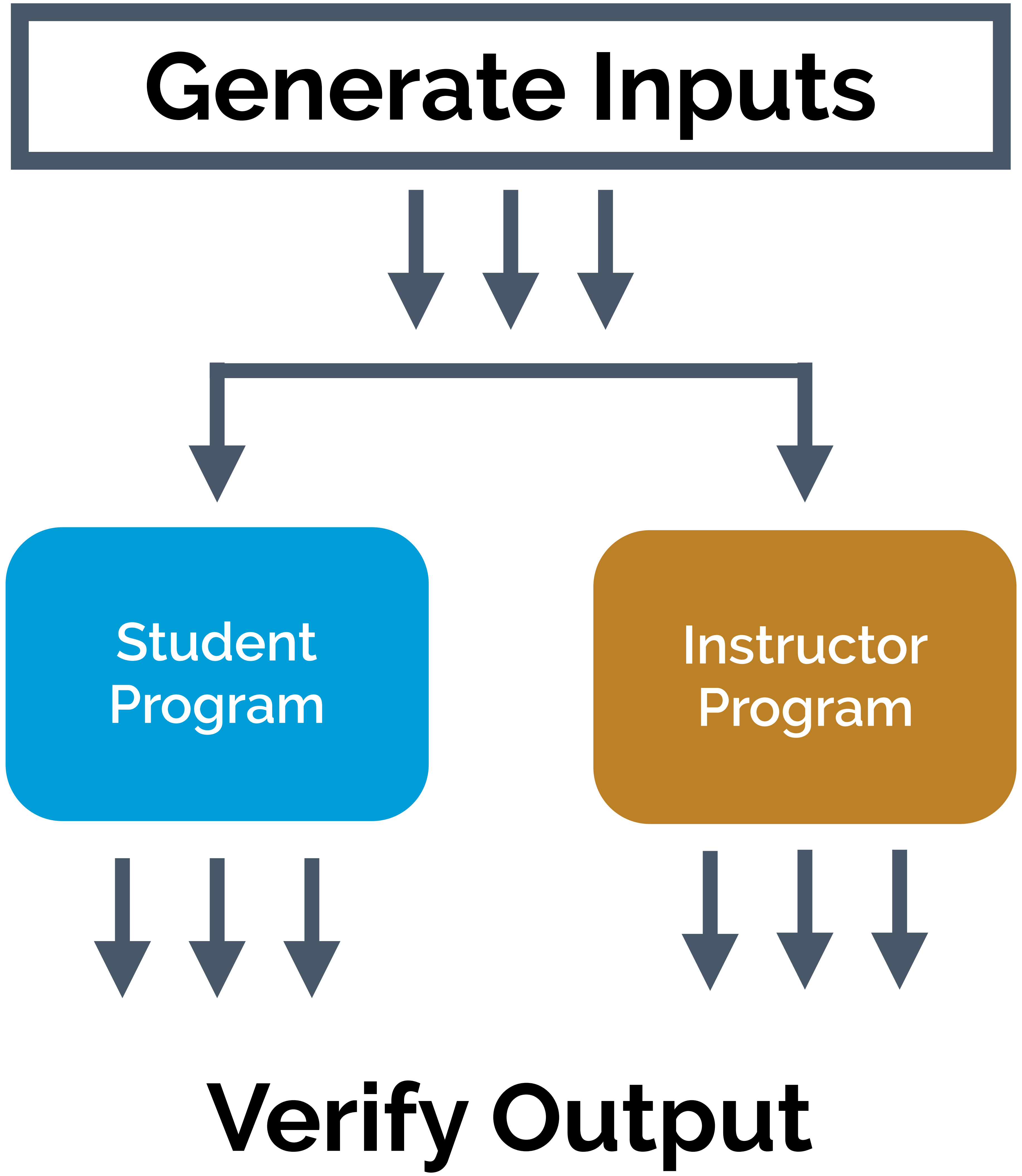
Expected Output

How to compare output?

How many inputs are required?

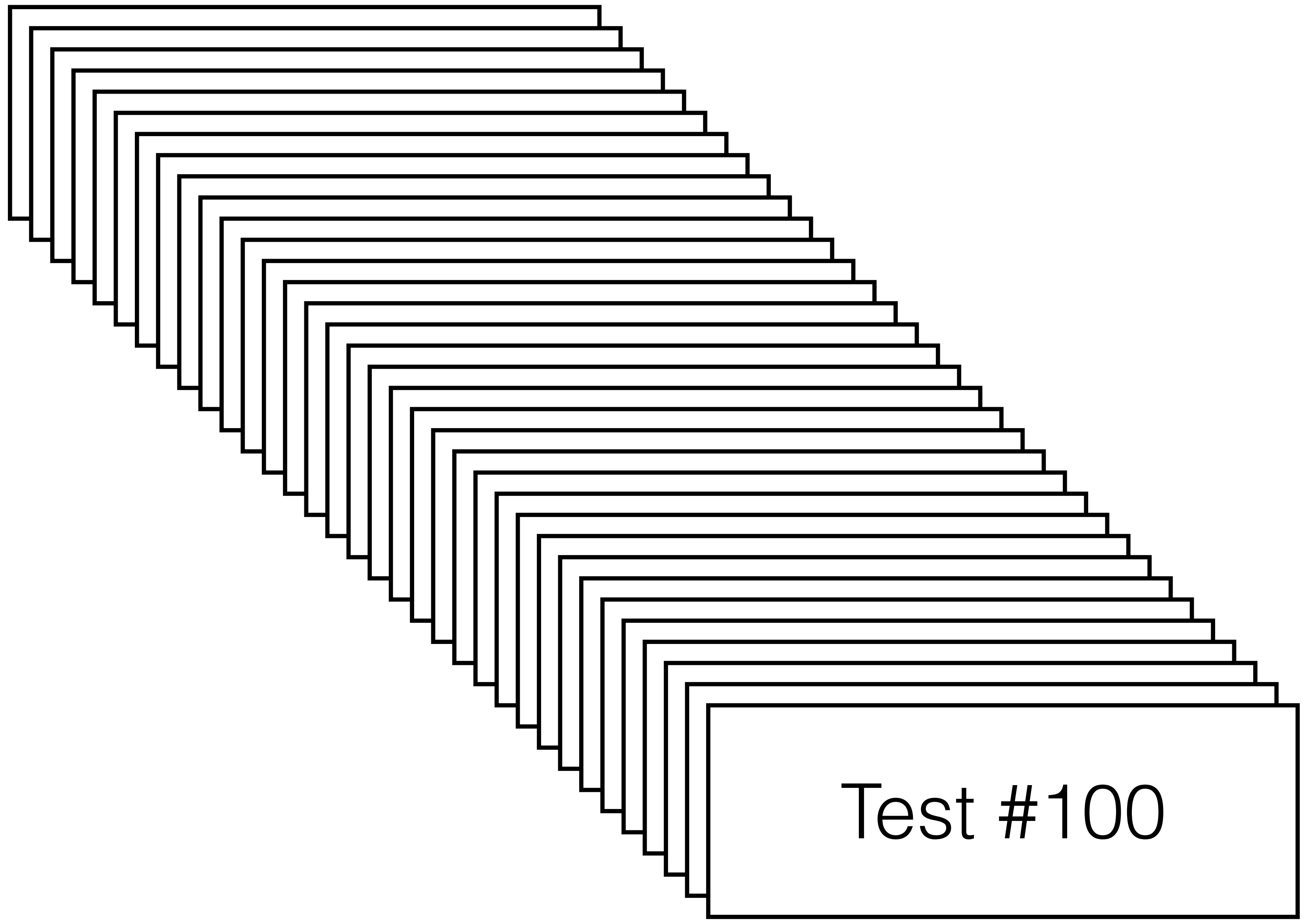
How to improve upon Fuzz Tests?

Creating & Distributing Tests



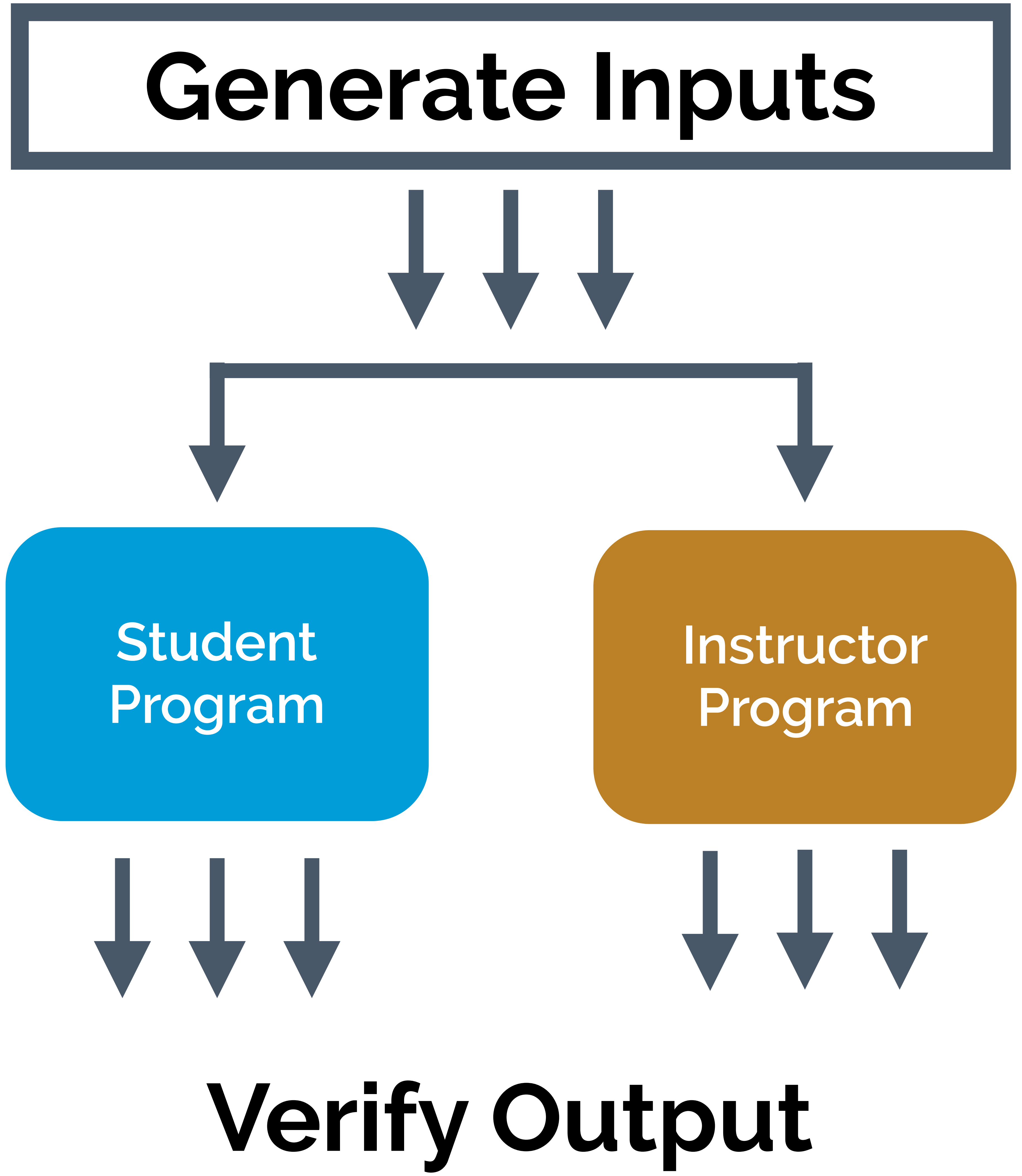
- Input domain is known
- Generating Random Inputs is easy

Generate Inputs



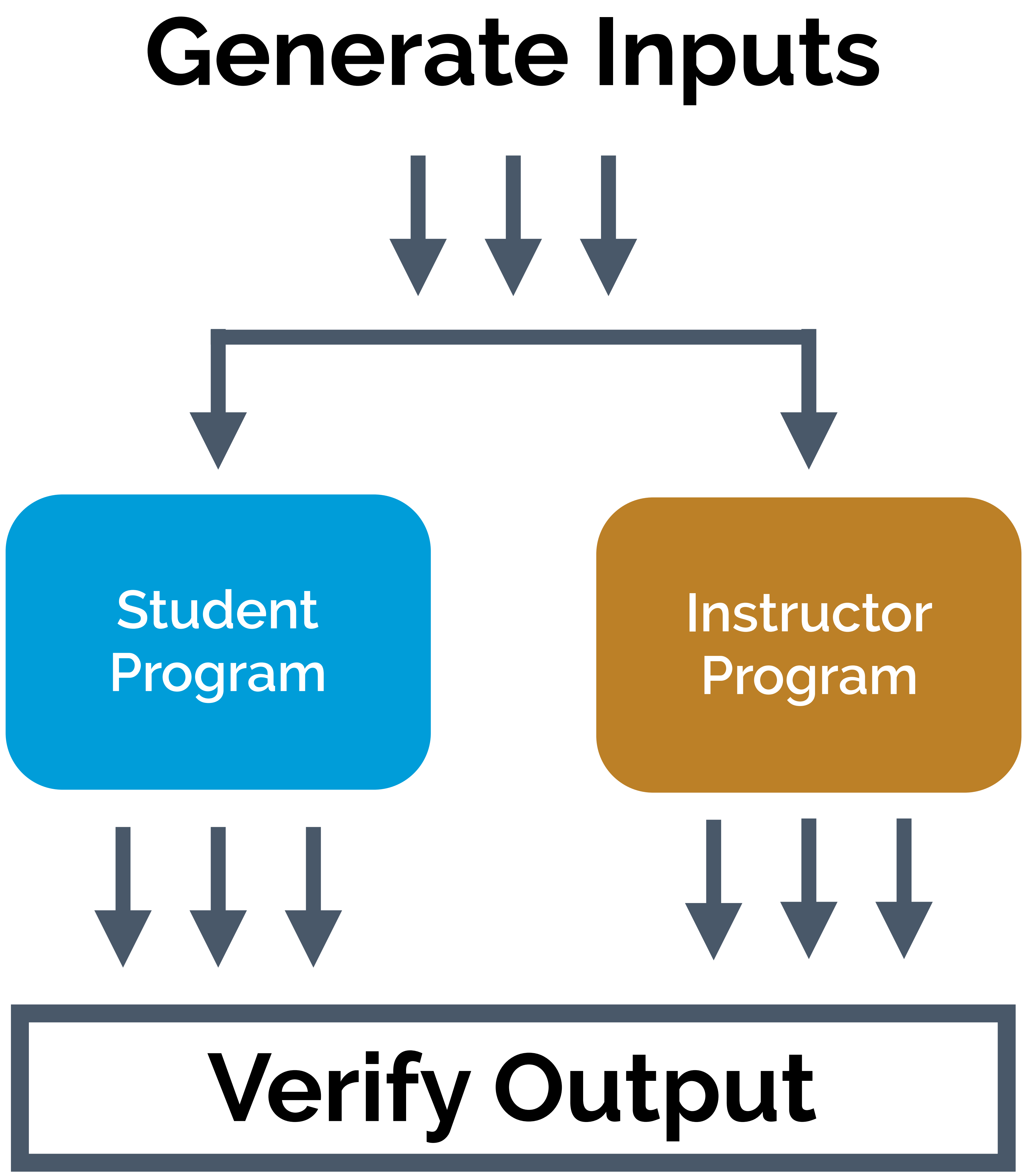
- Random input generation
- All students were provided with an identical set of tests
- Large number of inputs needed

Creating & Distributing Tests



- Input domain is known
- Generating Random Inputs is easy

Creating & Distributing Tests



Generate Inputs

Student Program

Instructor Program

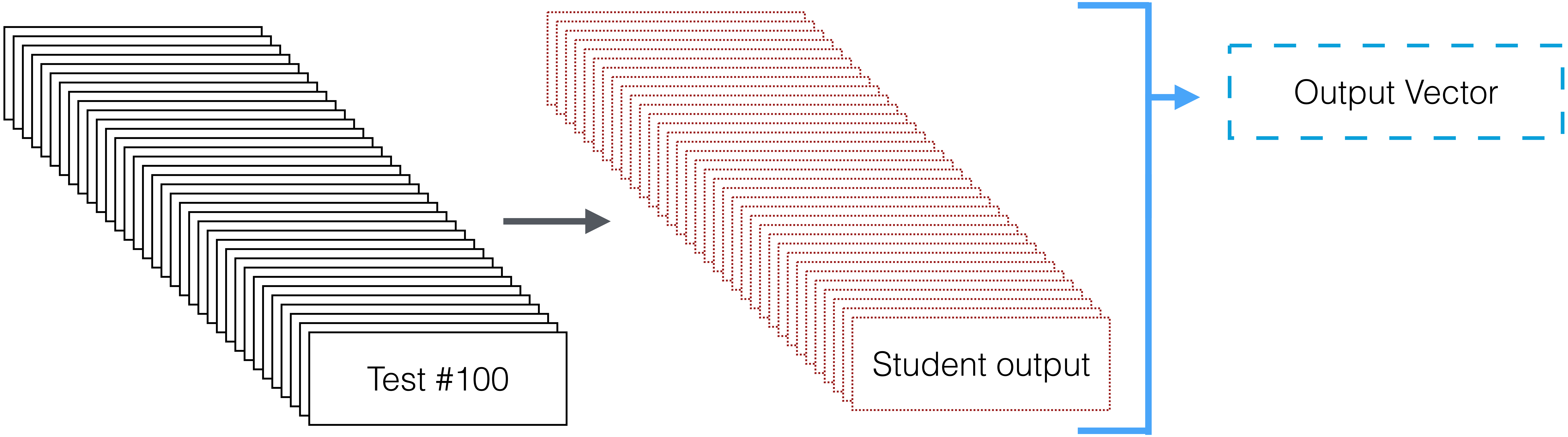
Verify Output

- Raw Output Check
- Hashing
- Program Tracing

RQ1: How to Compare Outputs?

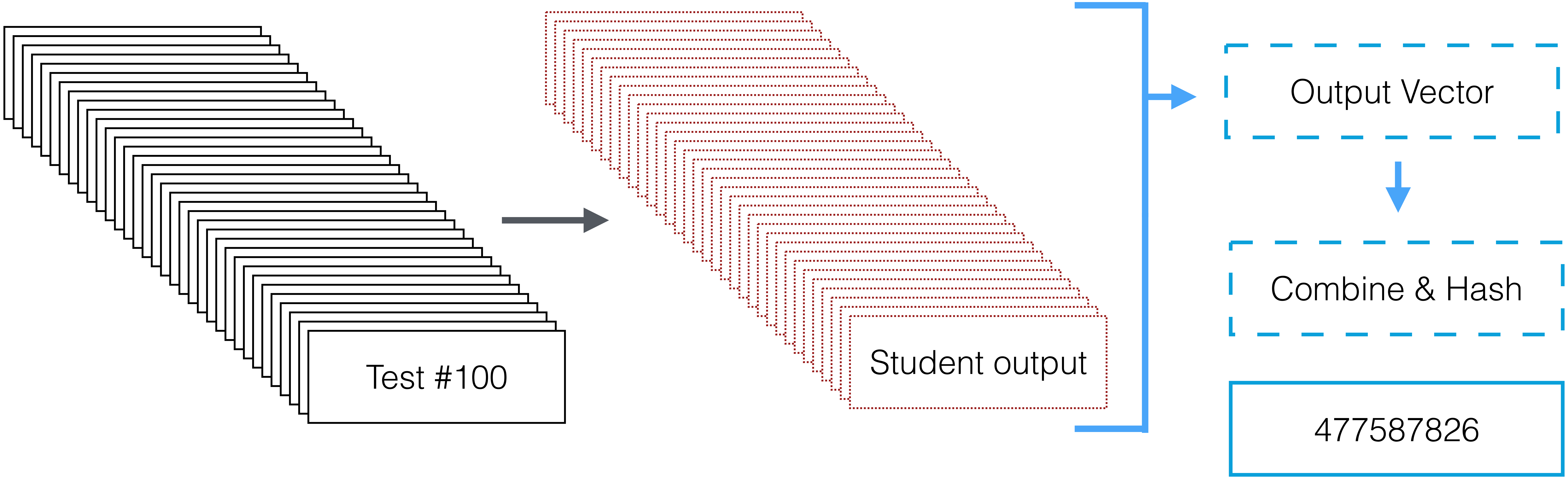
Verify Output

- Raw comparison of output
- Compare against precomputed result



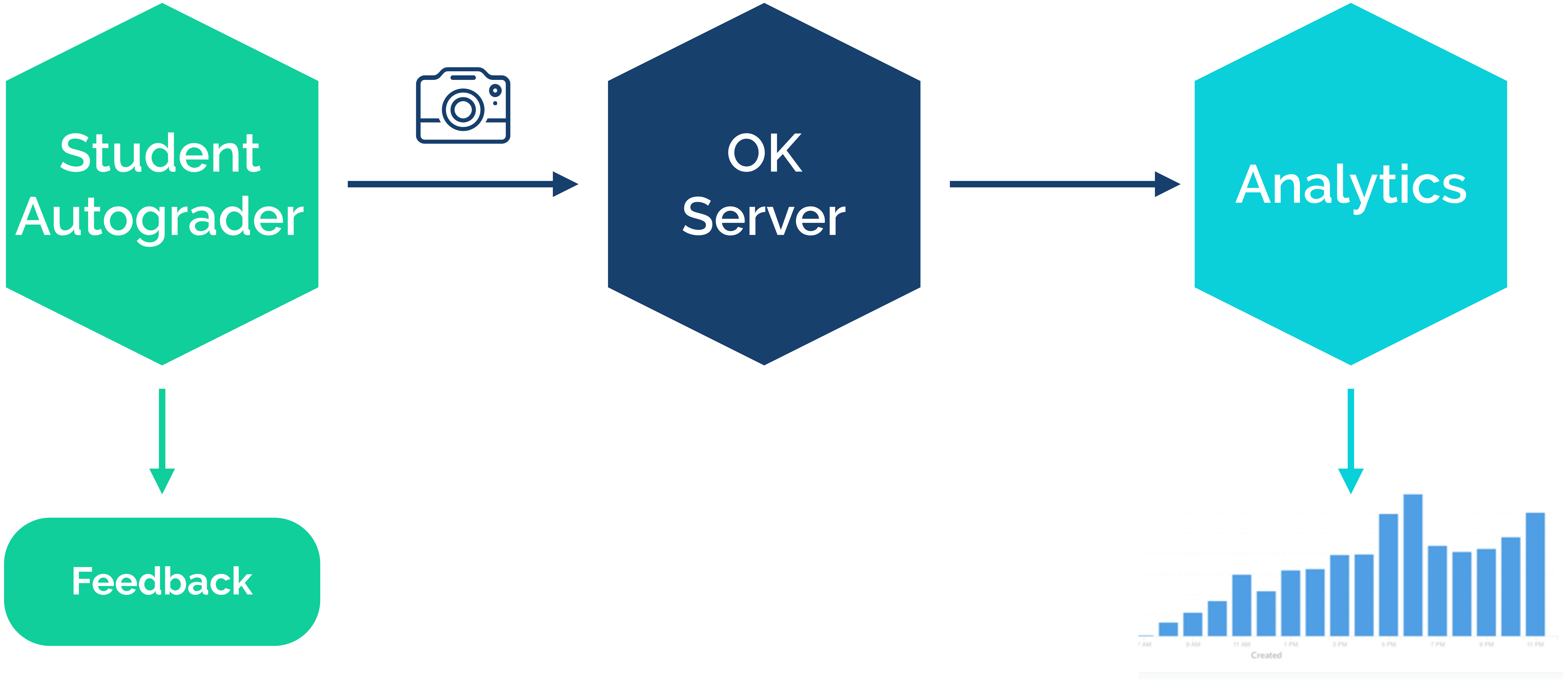
Hashing Output

- Hash combination of outputs
- Compare against precomputed result



CS61A @ UC Berkeley
cs61a.org

In Person CS1 Course with 1400 Students Enrolled



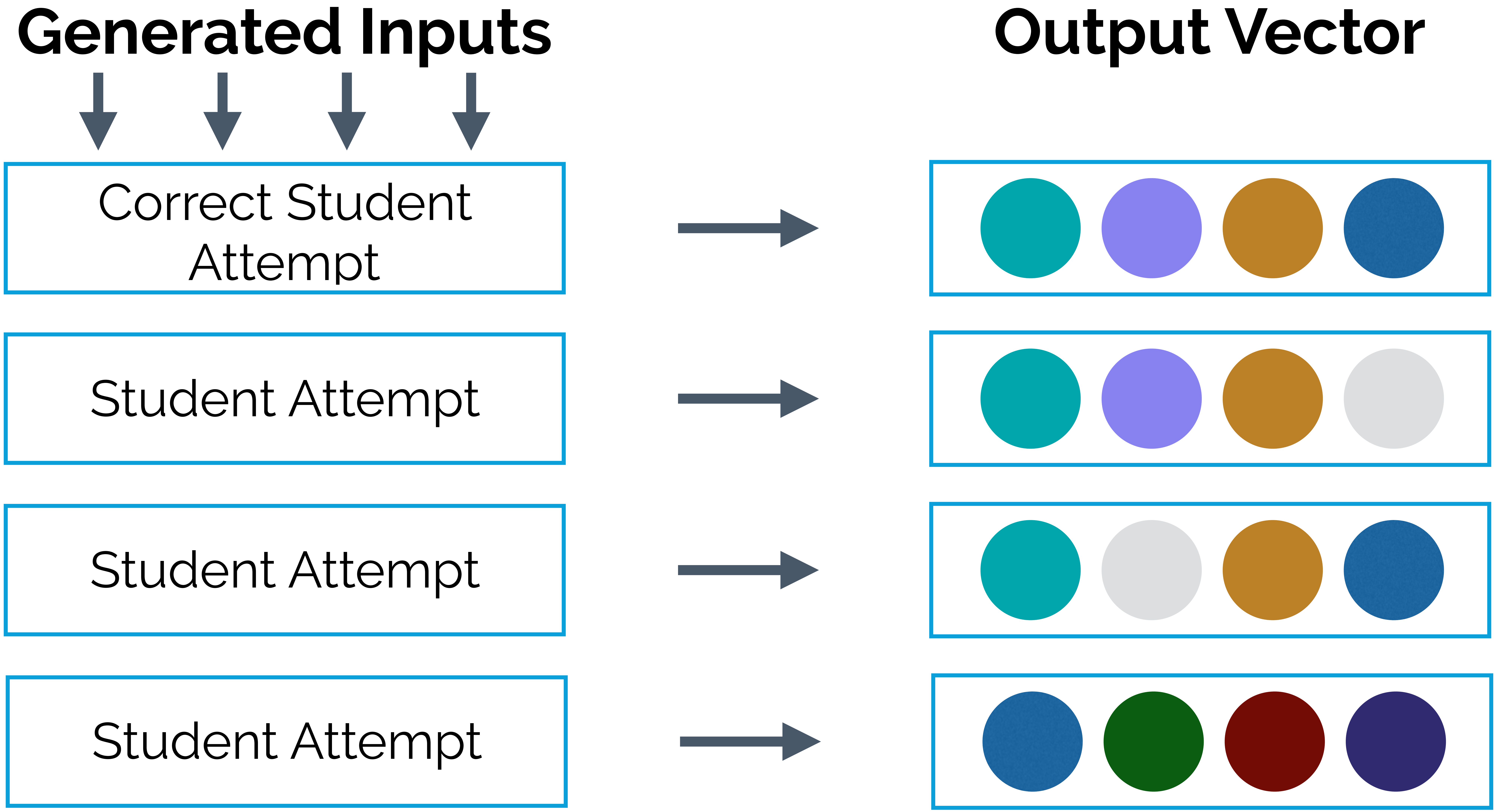
Student Autograder
OK Server
Analytics
Feedback
7 AM
9 AM
11 AM
1 PM
3 PM
5 PM
7 PM
9 PM
11 PM
Created

okpy.org

Collected Dataset

Students Completing Project	1,331
Code Snapshots	486,482
Average Snapshots per Student	349
Incorrect Attempts at Target Question	48,079

Output Vectors



Generated Inputs

Output Vector

Correct Student Attempt

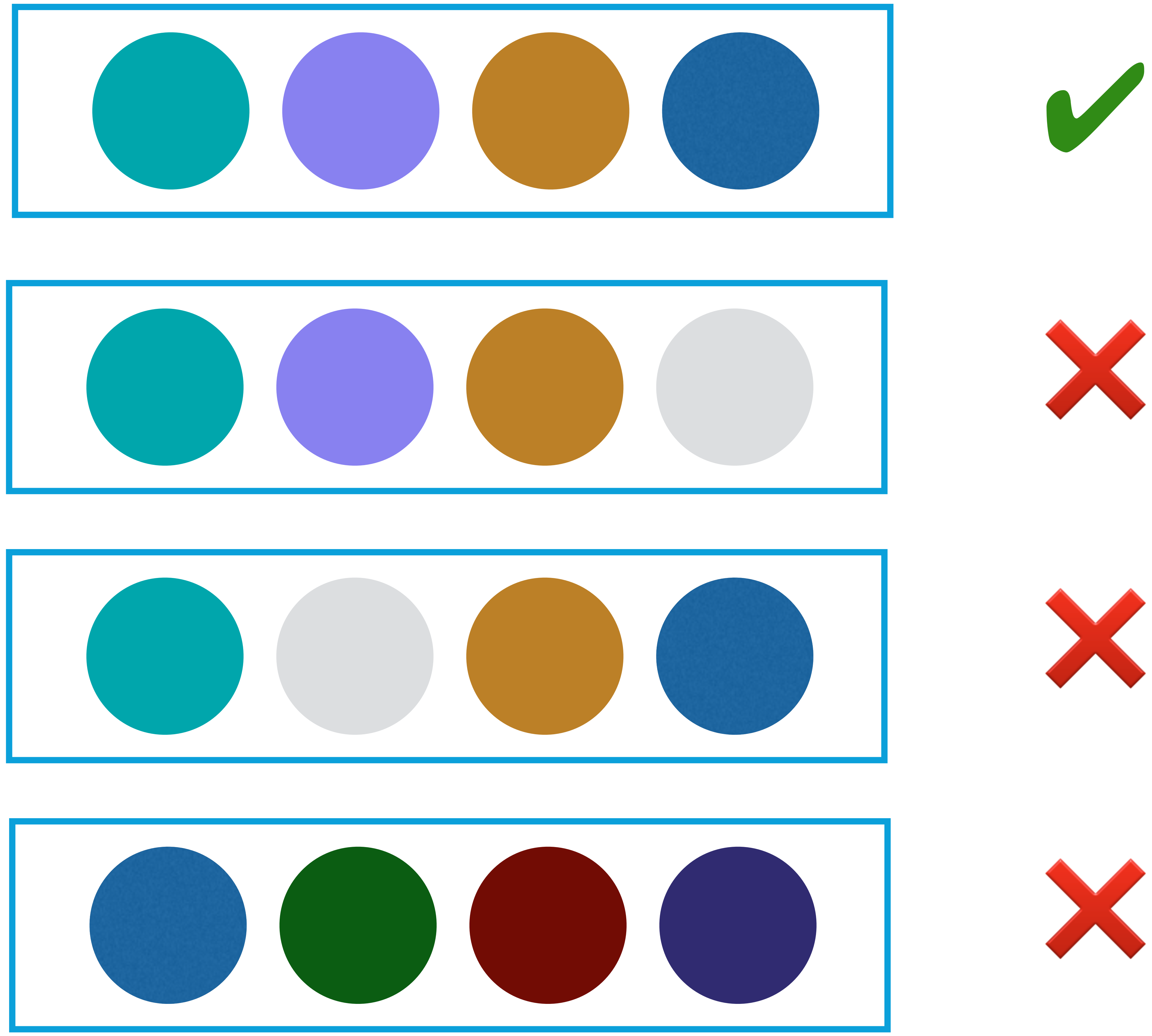
Student Attempt

Student Attempt

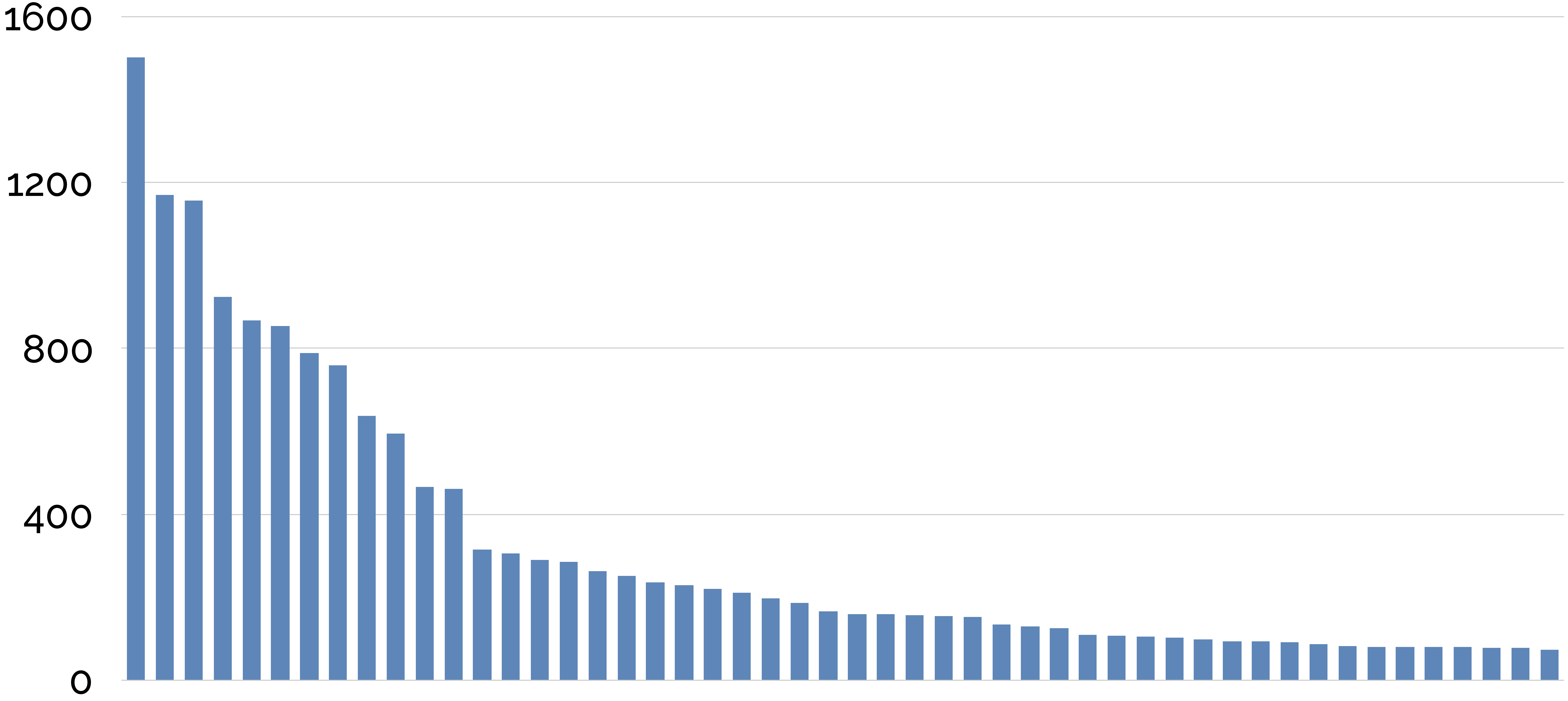
Student Attempt

Output Vectors

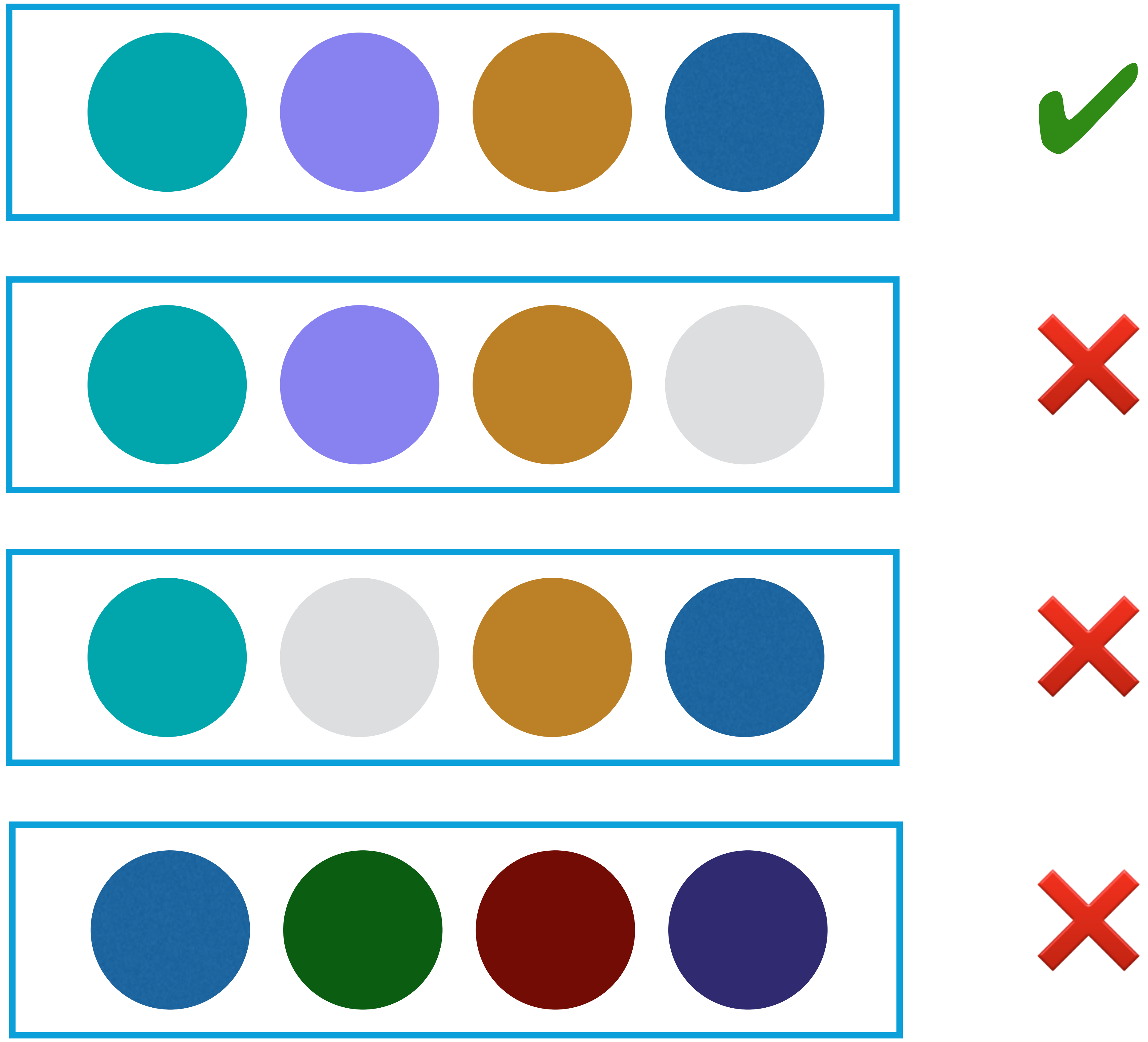
Output Vector



Frequency of Incorrect Outputs

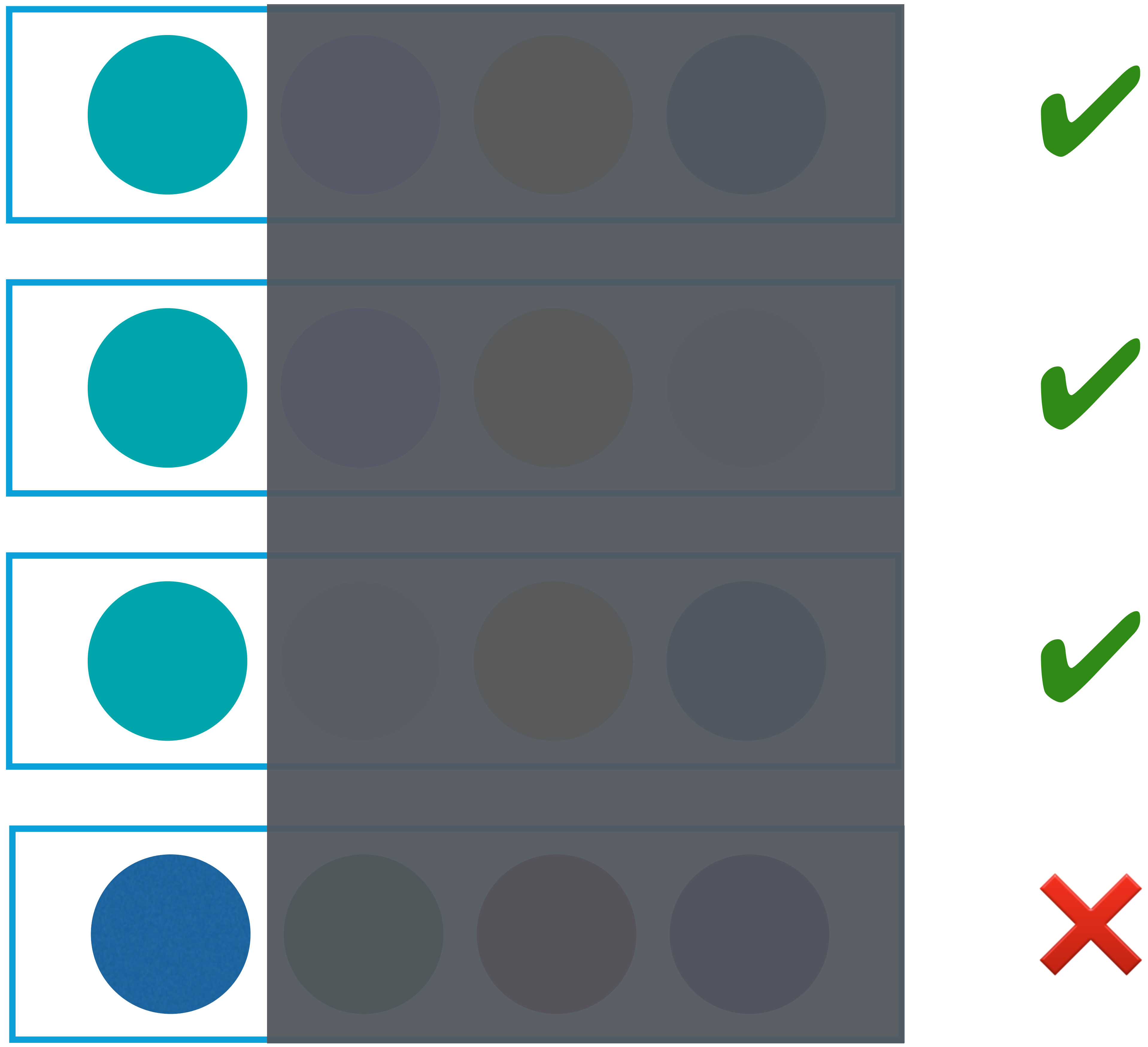


RQ2: How many inputs are needed?



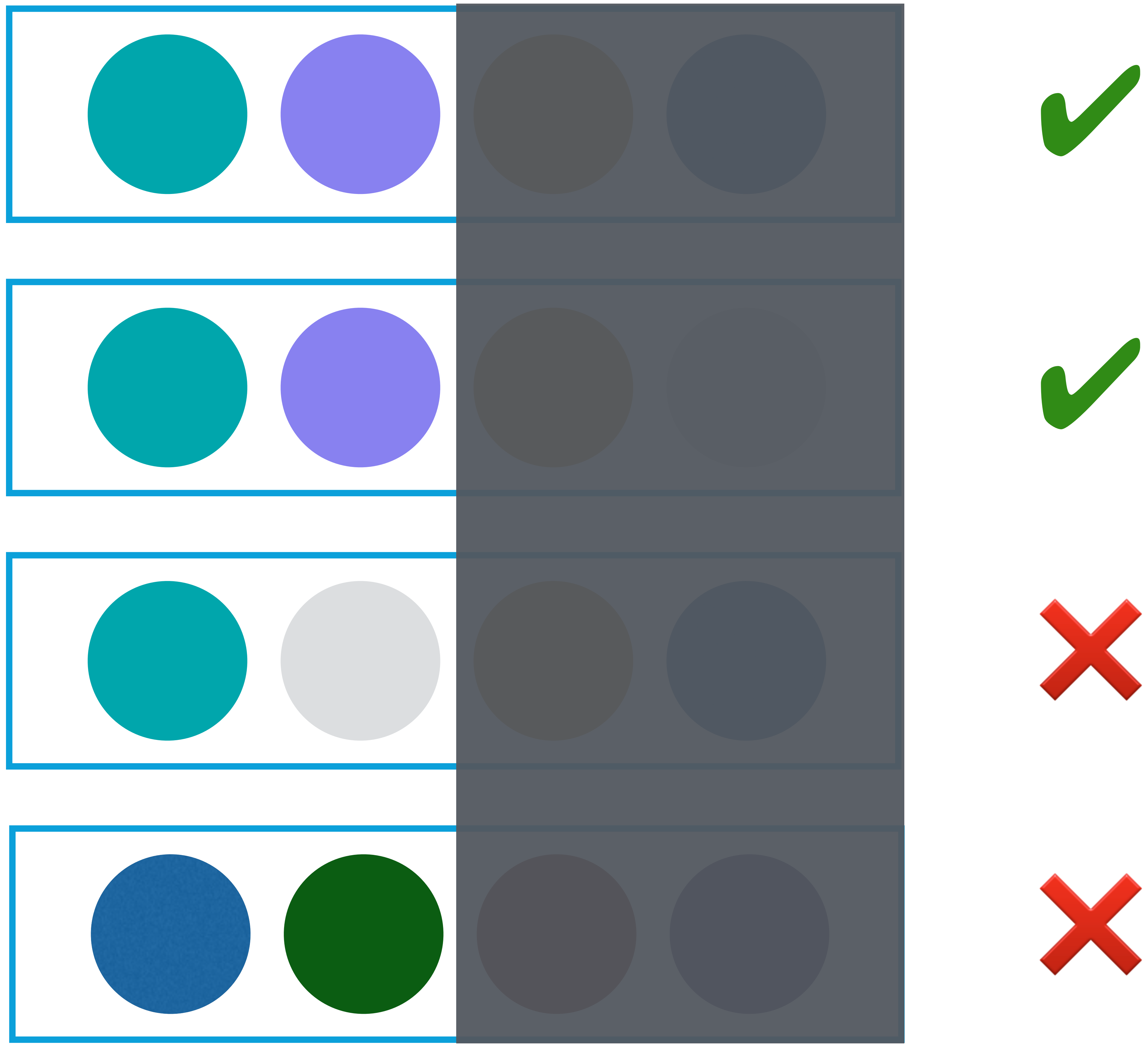
R2: How many inputs are needed?

1 Input



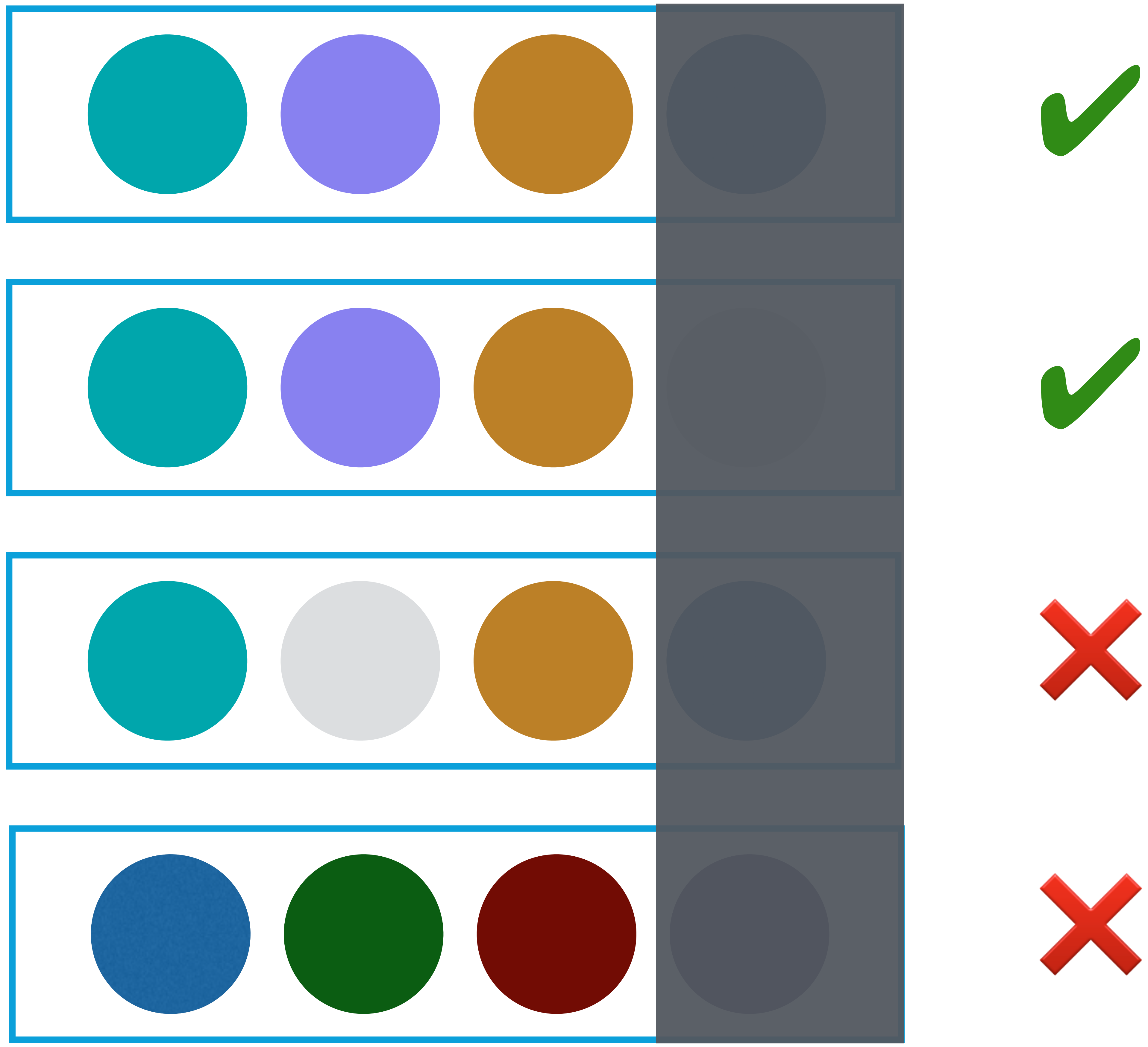
RQ2: How many inputs are needed?

2 Inputs



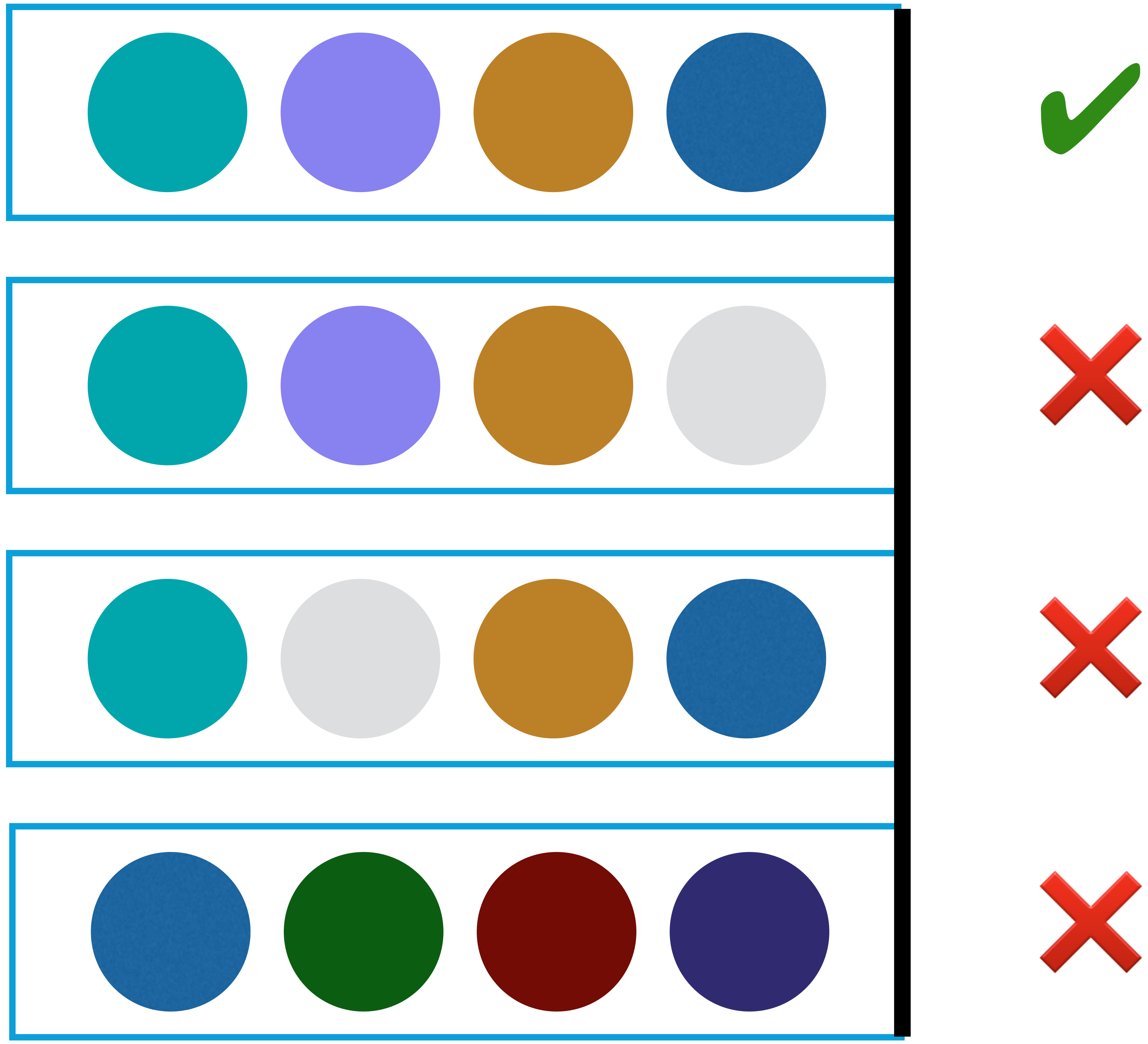
RQ2: How many inputs are needed?

3 Inputs

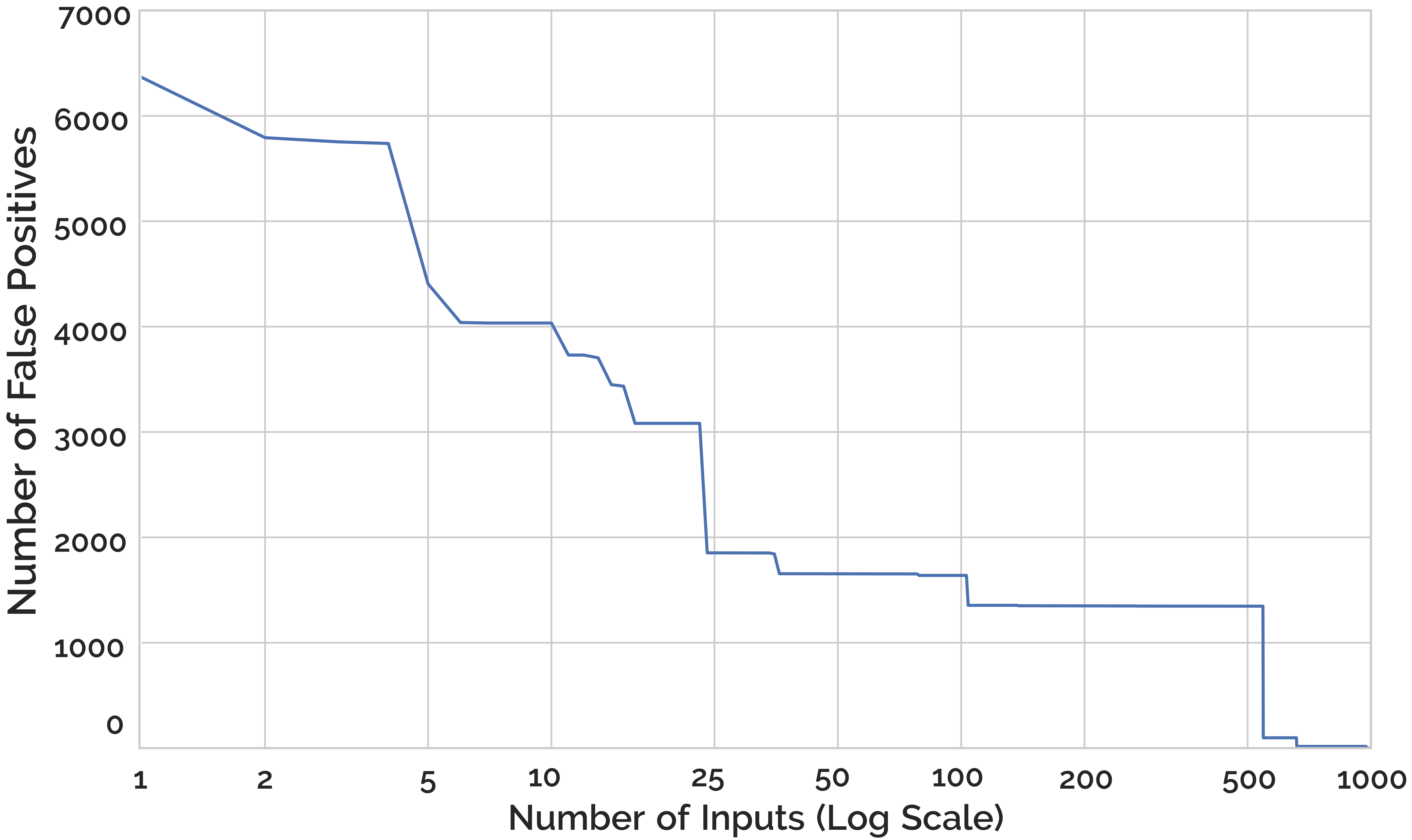


RQ2: How many inputs are needed?

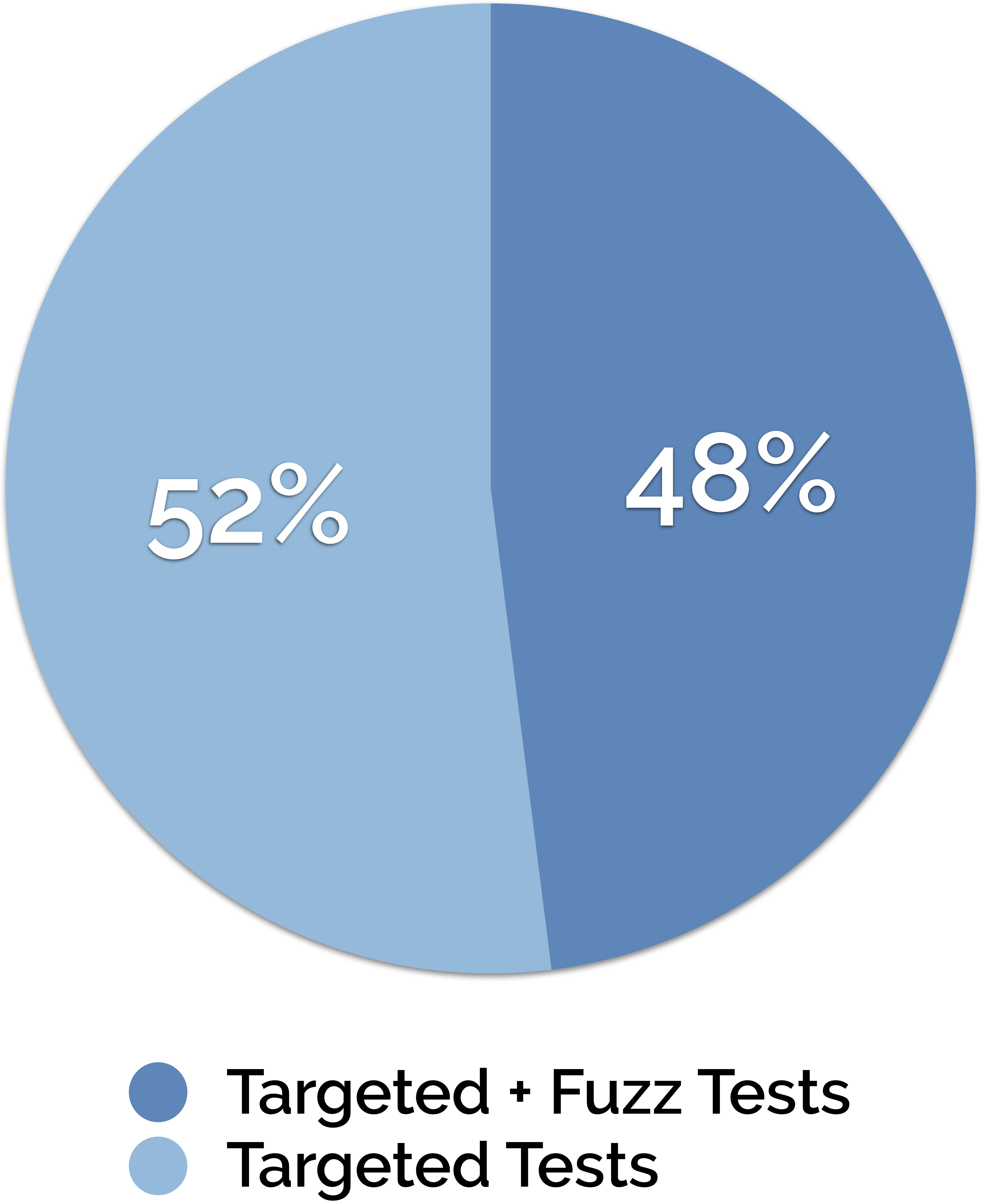
4 Inputs



RQ2: How many inputs are needed?

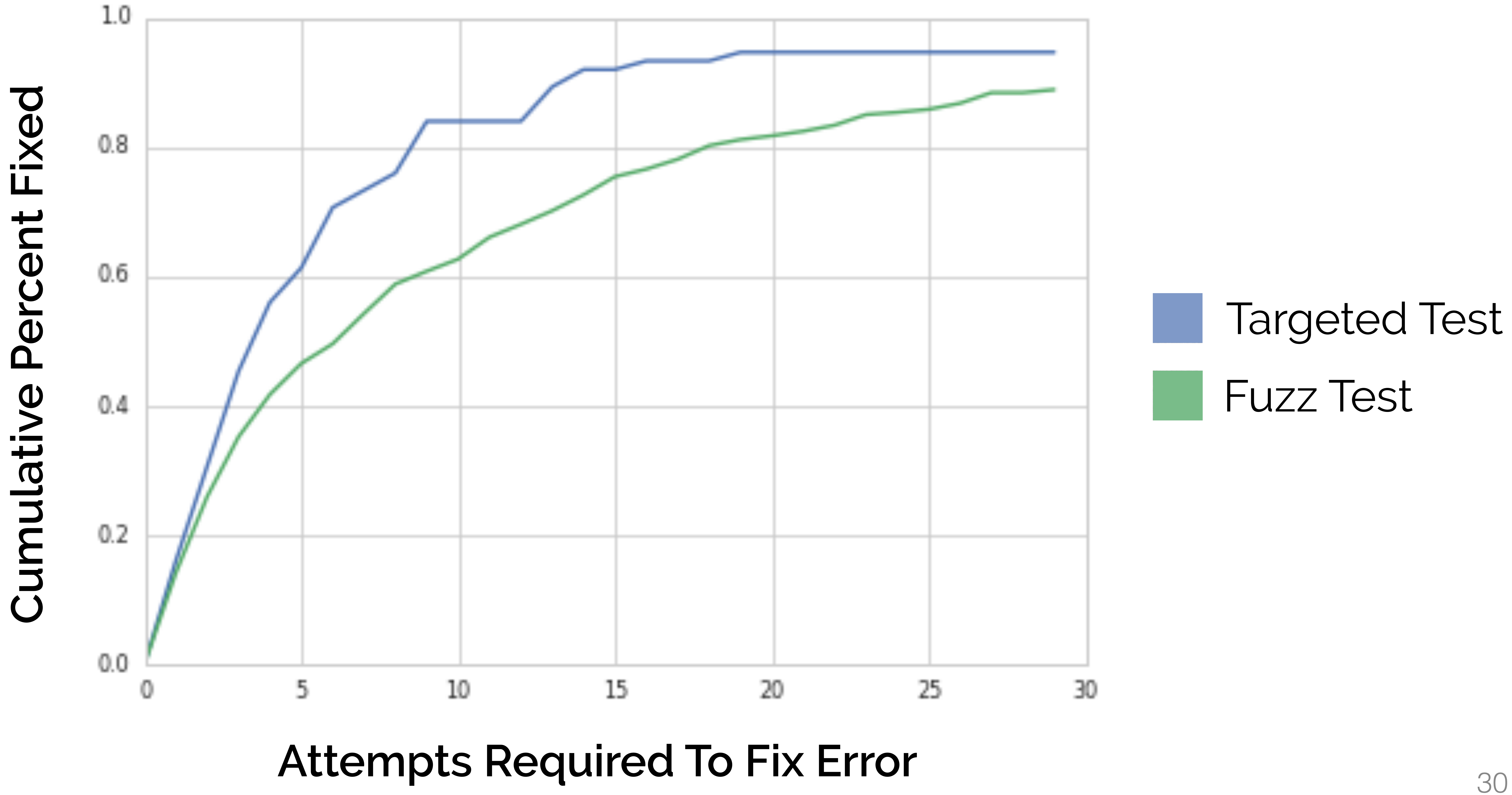


Fuzz Testing Effectiveness

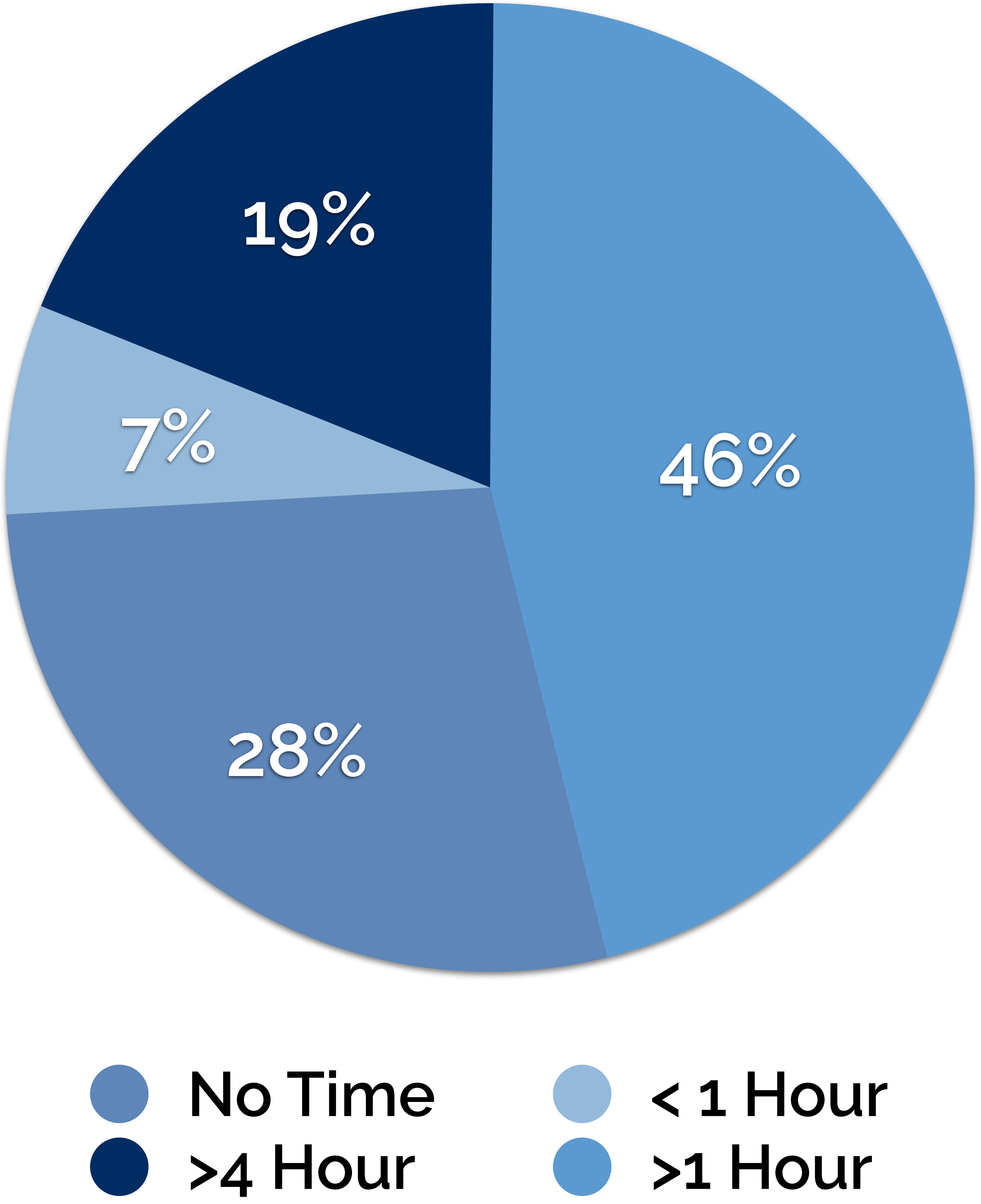


656 Students (48.4%) passed all of the targeted tests but still had an error caught by the Fuzz Tests

RQ3: Improving Fuzz Tests



How to Improve On Fuzz Tests?



As a result of the Fuzz Test:

46% of students reported spending 1 to 4 hours debugging

19% of students reported spending more than 4 hours debugging

Obfuscating output made it harder for instructors to help students

Program Inspection

```
Incorrect result after playing 1 game(s):
----------------------------------------------------------------
            score0  score1    Turn Summary
----------------------------------------------------------------
Turn 0:          0       0    Player 0 rolls 0 dice:
                +1
                 1       0
----------------------------------------------------------------
Turn 1:          1       0    Player 1 rolls 7 six-sided dice:
                       +37        3, 4, 6, 3, 3, 4, 6
                 1      37    Dice sum: 29
----------------------------------------------------------------
...
----------------------------------------------------------------
Incorrect implementation of game at turn 1.
Please read over the trace to find your error.
(error_id: 1189294328)
```

Thank you

sumukh@berkeley.edu

@sumukhsridhara

okpy.org

cs61a.org