

Automated Test Generation

Question 1. Consider the following class:

```
public class Lock {  
    private final static int UNLOCKED = 0;  
    private final static int LOCKED = 1;  
    private int state;  
    public Lock() { state = UNLOCKED; }  
    public void acquire() { assert(state == UNLOCKED); state = LOCKED; }  
    public void release() { assert(state == LOCKED); state = UNLOCKED; }  
    public int getState() { return state; }  
}
```

Consider each of the below five tests as a unit of input to Randoop. Answer whether that test can ever possibly be generated by Randoop. If yes, explain whether Randoop **1.** Discards the test as illegal, or **2.** Reports the test as a bug, or **3.** Adds the test to components for future extension.

For simplicity, assume that Randoop does not check for redundancy, and that the contracts it checks (not shown for brevity) are standard ones (e.g., equals and hashCode).

#	Test	Can ever be generated?	If yes, what outcome?
1.	Lock l = new Lock(); l.acquire(); l.release();	A. Yes B. No	1. Discards 2. Reports 3. Adds
2.	Lock l = new Lock(); int s = l.getState(); if (s == 0) l.acquire();	A. Yes B. No	1. Discards 2. Reports 3. Adds
3.	Lock l = new Lock(); l.release();	A. Yes B. No	1. Discards 2. Reports 3. Adds
4.	Lock l = new Lock(); l.release(); l.acquire();	A. Yes B. No	1. Discards 2. Reports 3. Adds
5.	Lock l = new Lock(); l.acquire(); l.release(); l.acquire();	A. Yes B. No	1. Discards 2. Reports 3. Adds