**CS499: THE ANNOTATED MIDTERM**

**MEDIAN: 90**
**MEAN: 89**

---

1. Which of the following is NOT often a characteristic of conventional embedded software?

a. A sophisticated GUI is a primary component of the system
b. Written in a low-level language such as C or C++
c. Involves computation connecting sensors and actuators
d. More critical that it behaves well than for typical application software

Literally nobody missed this. I think I can pass to the next question.

---

2. Which of the following is a true statement?

a. libFuzzer is often faster than AFL, due to its focus on fuzzing functions rather than executables.
b. Coverage-driven fuzzers never require re-compiling a program.
c. There is a single, best fuzzer that almost always outperforms other approaches.
d. Fuzzers like AFL generate new inputs from scratch for every test, rather than adapting previously discovered or provided inputs.

22 (of 31) of you got the right answer, (a). 8 people answered (d), but AFL and so forth work precisely by modifying inputs they already have in the queue. 1 person put (b) which would be nice, but no such luck.

---

3. Failed test/crash triage is

a. ususally easy, because fuzzers only discover a small number of bugs, and distinguishing them is always simple.
b. a manual process in which no automatic assistance is possible.
c. an important part of the full process of discovering, reporting, or fixing bugs with fuzzing.
d. only possible in languages such as Java that support garbage collection.

Almost everyone chose, correctly, (c); one person put (a) and that would also be nice but is not true.

---

4. Simulating hardware faults is

a. best done by actually modifying hardware.
b. an important part of file system testing.
c. not important, outside of the most mission-critical systems such as medical devices or nuclear power plants.
d. most commonly performed by modifying AFL or another fuzzer.

Really surprised here; 22 of you picked (b), but 4 picked (a) – which is really hard and would be slow, 4 picked (d) – which I have no idea how that would even work and 2 picked (c), again not true. You'd like your laptop not to lose the file system every time you run out of battery, for example.

---

5. The choice of programming language in a system under test

a. is of no interest testers, vs. developers.
b. is ususally a functional language such as Lisp or Haskell, for embedded systems.
c. is important because it influences the kinds of bugs that will be most prevalent.
d. is usually determined after testing is complete.

We had 25 (correct) (c) answers, and 4 (b) answers (the opposite is true: nobody writes embedded Haskell or Lisp, really), plus 2 (a) answers. But (c) is why (b) is wrong – language influences bug types and fuzzing tools available.

---

6. Test case reduction is

a. always based on complex machine-learning algorithms using neural nets.
b. currently a completely manual activity.
c. based on systematically trying to remove or change parts of a test.
d. not particularly useful to testers, vs. developers.

28 people correctly chose (c); 2 picked (b) – but there are tools here, in fact purely manual reduction is very seldom done (too hard). 1 person chose (a) which is just not true. I think I haven't even mentioned neural nets.

---

7. DeepState is

a. based on the idea of generalizing or parameterizing unit tests.
b. used for coordinating testing across docker containers, via Amazon APIs.
c. written primarily in Java.
d. not useful for embedded and low-level code.

Again, 28 correct (a) answers. 2 people chose (b) because we mentioned docker a lot, probably. 1 person chose (c) and I hope it was a joke!

---

8. Sanitizers

a. add to the set of circumstances under which code will crash.
b. slow down fuzzing.
c. sometimes produce a hard-to-manage amount of output.
d. all of the above

Almost everyone got (d) here; there was one (a) which is true, but so are (b) and (c).

---

9. Two sanitizers we discussed in class are:

a. address sanitizer and database sanitizer.
b. address sanitizer and undefined behavior sanitizer.
c. memory sanitizer and cloud sanitizer.
d. hand sanitizer and logic sanitizer.

29 of you answered (b) correctly; 2 people somehow thought three was a database sanitizer, which sounds neat but I have no idea what it would do.

---

10. Which of the following is true?

a. DeepState is a closed-source, proprietary system only available under an expensive license.
b. AFL is better than libFuzzer.
c. libFuzzer is better than AFL.
d. It is easy to construct small programs most or even perhaps all current fuzzer cannot easily generate a crashing input, despite the fact that one exists and can easily be determined by a human.

Again, nobody missed this!