1 Delta Debugging (13 points)

The Delta Debugging (DD) algorithm makes certain assumptions about its input. It requires that the interesting function be *monotonic*, *unambiguous* and *consistent*. Consider the following snippet of code.

```
1  int macaroni (int s) {
2      int *x = NULL;
3
4      x = (int*)  malloc (s * 1024 * 1024 * 1024 * sizeof(int));
5
6      if ( x == NULL ) {
7          return -1;
8      } else {
9          return 0;
10     }
11 }
```

Say we want to minimize a test suite for this macaroni function.
malloc is a function that asks the system to allocate the specified amount of memory — it returns a pointer if allocation is successful, and NULL if the memory cannot be allocated. (Reminder: 1024 bytes = 1KB, 1024KB = 1MB, etc. Also, what does it mean to allocate negative bytes?).

(a) *(2pts. total)* Assuming your development system has 16GB of memory available to respond to malloc requests, 32-bit integers, that this program is run in complete isolation, AND memory allocation is perfectly optimal, complete the table of test cases and corresponding output.

Test #	**Input (int s)**	**Output of macaroni?**
0	2	F
1	48	T
2	3	F
3	-1	T
4	-3	T

(b) *(3pts.)* If we want to minimize this test suite with respect to line coverage, how would you define the interesting function?

Interesting would be defined as whether test suite coverage lowers. The set becomes interesting if the coverage is reduced. This parallels HW5.

(c) *(5pts.)* Below, write the numbers of the test cases that minimize the test suite with respect to line coverage.

Answers will vary, but should consist of 2 inputs: 1 with a small positive integer (like 2 or 3) and 1 with a larger or negative integer.

(d) *(3pts.)* Now, ignore the assumptions from (a). In two sentences or fewer, explain which Delta Debugging assumption is violated by the code above, and why. Hint: keep in mind the behavior of malloc as described above.

Because we may not be certain about the computing environment, it could be the case that malloc fails on an input that would otherwise be normal. For instance, if you run the test suite on a highly-utilized computer with many concurrent programs, you run the risk that malloc fails. This could violate unambiguity in that Test 1 with input X succeeds, but Test 2 with input X fails. You could also argue for consistency if you redefine how interestingness is defined (e.g., "five out of five runs are interesting otherwise, it is unknown."

2 Design Patterns (14 points)

Consider a multiplayer computer game. In this game, each player controls the movement of their character. Whenever a player moves their character, all other players see where that player's character moves. Your tragic developer provides the following Java code.

```
1  class Point {
2      public int x, y;
3      public Point (int _x, int _y) {x=_x; y=_y;}
4  }
5  class Game {
6    Player players[32];  // 32 players
7    public startGame() {
8      for (int i=0; i<32; i++)
9        this->players[i]  = new Player();
10
11     while (true) {
12       // main game loop
13       for (int i=0; i<32; i++)
14         players[i].move( random(), random() );
15
16       for (int i=0; i<32; i++) {
17         for (int j=0; j<32; j++)  {
18           if (i==j) continue;
19           players[i].updateOtherPlayer(players[j]);
20         }
21       }
22     }
23   }
24 }
25 class Player {
26   public Point pos; // position
27   public Player () {
28     // constructor
29     pos = new Point(0,0);
30   }
31   public void move( int _x, int _y ) {
32     // move player to new position
33     pos = new Point(_x, _y);
34   }
35   public Point updateOtherPlayer (Player p) {
36     // update information about another player
37     Point otherPos =  p.pos;
38     this.updateGameScreen( p, p.pos);
39   }
40 }
```

You may make the following assumptions about the code.

- `random` is a helper method that returns a random integer.
- `updateGameScreen` is a helper method that updates one player's view of another individual Player's position.

(a) (*6pts.*) Identify two (2) design patterns that might be useful in this program and explain why. (One sentence each).

Publish-subscribe – multiple players waiting for information from a server; Singleton – one instance of game. You could also argue for MVC (with discussion about the Player's view and the Game instance modeling player motion) or factory (e.g., with the Point class)

(b) (*6pts.*) Suppose you want to reorganize the `Game.main` method. Further suppose that you add a method to `Game`:

```
1 ... (other stuff from Game) ...
2 public void updateAllPlayers( Player p ) {
3   for (int i=0; i<32; i++) {
4      if (players[i] == p) continue;
5      this.players[i].updateOtherPlayer( p );
6   }
7 }
8 ... (other stuff) ...
```

If you wanted to call this method from `Player.move`, what change would you need to make to `Player.move`? (One sentence or less).

You need to couple the Player class with Game by passing in a Game instance to the method as well.

(c) (*2pts.*) In one word, what property are you increasing by making the change above?

Coupling. You're increasing coupling between Game and Player.

3 Short Answer (21 points)

(a) (*3pts.*) During Chad Spensky's guest lecture, he placed heavy importance on keeping projects open source. Explain why in two sentences or fewer.

Spensky's argument was that it makes you a better developer. Open sourcing your own work holds you accountable for making software better with respect to non-functional properties.

(b) (*2pts.*) In two sentences or fewer, describe the role of the requirements engineer during elicitation.

No partial credit. The requirements engineer is tasked with eliciting the requirements from the customer via stakeholder analysis, domain knowledge research, and interviews.

(c) (*3pts.*) In Jack Wadden's guest lecture, he explained a scenario in which a difference in a simulation versus real hardware led to a difficult defect to track down. Explain how Delta Debugging might be used to help discern such a difference in compilation tools.

Delta Debugging is use for test suite minimization. Wadden could have used DD to help develop a test suite in which interestingness was defined as a difference between simulated and physical output. A minimal test suite could have been used to find specific locations in Verilog code that led to the difference in behavior.

Half credit for mentioning minimization, half credit for mentioning interestingness.

(d) (*2pts.*) Chad Spensky explained that it was important to "pro up" with various software engineering tools. Describe what this means in one sentence.

This is an informal argument that developers should become experts at the tools they use. Also calling back to the productivity lecture, expertise in specific tools can help you get work done faster.

(e) (*3pts.*) Explain in your own words a specific scenario that might involve a multi-language project. Use two sentences or fewer.

Answers will vary. Projects can vary from things like TensorFlow to personal projects. 1 point for vague answer (e.g., a project that needs a fast C kernel), 2 points for a description of a project that involves multiple languages but no cross-language interaction, and full credit for a project that describes in detail a project where multiple languages interact.

(f) (*5pts.*) You are interviewing a customer to gather requirements. They are interested in a 2D graphics program that draws specified polygonal shapes on the screen. They indicate a specific screen resolution and color depth, and say that no more than 3 shapes with 5 sides or fewer each will be drawn on the screen at once, that all shapes are regular (all angles are equal in each shape), and that the shapes must be drawn in under 1 second. You deliver a prototype that you think meets the requirements — however, the customer says that the graphics do not look right.

Specify (1) what might have gone wrong in the prototype you delivered that led the customer to be unhappy, and (2) what might have been done during elicitation to avoid such mistakes from occurring. Use four or fewer sentences.

Answers can vary.

Customer and engineer may have miscommunicated whether the shapes were solid, or the extent to which the shapes could overlap.

Interview could be improved by preparing prototypes ahead of the customer interview, or by substantial followup questioning to determine further details about the customer's needs.

(g) (*3pts.*) In class, we discussed how activity in the brain changes based on expertise. Support or refute the claim that measured neural efficiency should be used as a basis for evaluating a software engineering candidate.

Support: neural efficiency is a proxy for expertise in other domains – using it can help objectively identify software engineering expertise. Refute: aside from ethical (we don't want to be like Gattaca) and practical (it may be too costly) concerns, neural efficiency from one person to another may vary and not be comparable.

4 Software Engineering Narrative (15 points)

(1 pt. each) Read the following narrative. Fill in each ____ blank with the single *most specific or appropriate* corresponding concept from the answer bank. (Each ____ blank does have *exactly one* corresponding answer.) Each option *can* be used more than once.

A. Conditional Breakpoint	B. Creational Design Pattern	C. Defect	D. Fault
E. Fault Localization	F. Feature Request	G. Priority	H. Profiling
I. Requirements Elicitation	J. Singleton Design Pattern	K. Stakeholder	L. Swiss Cheese Model
M. Traceability	N. Triage	O. Watchpoint	

(a) F The developers of "Bashing Mealworms" want to add support for different colored in-game Mealworm characters in response to player feedback.

(b) M The music software VeggieLoops has an individual test case for each individual requirement.

(c) N A defect report related to the Pizza topping "anchovies" is received. However, the report is closed because it is deemed not reproducible.

(d) D Your code compiles just fine, but running it against one test case seems to cause accessing memory beyond the boundary of an array, leading to a crash *at runtime*.

(e) J The Pizza company Trionomos wants to keep track of their statistics, such as pizzas sold, revenue to date, and number of locations in one centralized location.

(f) L A credit reporting company NutriFax experiences a massive data breach. Upon investigation, it was discovered that multiple failures occurred in sequence: the database did not throttle queries, the front-end allowed requesting millions of records at once, and the authentication mechanism allowed administrative access without valid credentials.

(g) E A developer on your team ranks lines of code by suspiciousness to identify a buggy "if" statement in their code.

(h) G In the middle of a sprint, your manager tells you to fix a bug identified in a defect report as soon as possible.

(i) C Your new intern inserts a null pointer dereference in the middle of your team's module.

(j) H You insert time measurements at the beginning of each function to determine which functions of code to optimize by reimplementing in native C.

(k) K The printer company Cannon is considering developing software that optimally conserves ink in their consumer printers. However, the CEO is concerned that this will overly diminish ink cartridge sales, and so cancels the project. The developers on the customer success team were very surprised by this move, given the interests expressed by customers.

(l) A A developer wants to stop a program's execution when the variable `x` equals 5.

(m) I Hannah approaches a new customer concerning their request for new pharmacy management software. Hannah prepares for interactions with this customer by reading

about prescription laws to determine the scope and costs of a pharmacy-related software project.

(n) O A developer wants to stop execution to step through instructions every time the address `0xdeadbeef` is written to.

(o) B The video streaming site BiliKan uses a single Video class to encapsulate manipulation of multiple video formats. Each video format has an associated subclass that works with that specific video format.

5 Interviewing (17 points)

You are a hiring manager considering three candidates. You have given each candidate the following prompt:
Given an array of integers of size $N+1$, each integer from 1 to N is in the array *once* except for one integer, which appears twice. Find the value of the integer that appears twice. If no pair exists, report an error.

(a) (*4pts.*) List 4 test cases that you would expect a good candidate to provide, and why.

Answers can vary depending on generated interviewee code. We would expect to see "normal cases" as well as corner cases (e.g., empty list, empty output, errant spaces, etc.)

Candidate 1 provides the following code.

```
def findDuplicate(nums):
    for i in range(len(nums)):
        for j in range(len(nums)):
            # consider each pair of numbers
            if nums[j] == nums[i] and i != j:
                # bail on the first match
                return nums[i]
    return None
```

Candidate 2 provides the following code.

```
def findDuplicate(nums):
    nums_unique = set()
    for i in nums:
        if i not in nums_unique:
            nums.add(i)
        else:
            return i
    raise Exception('No pair')
```

Candidate 3 provides the following code.

```
def findDuplicate(nums):
    nums.sorted()
    for i in range(len(nums)):
        if i == 0:
            continue
```

```
6           if nums[i] == nums[i - 1]:
7               return nums[i]
8       return None
```

(b) (*10pts.*) In the table below, mark with a X the candidate that performed the best with respect to each property listed and explain why briefly.

	C 1	**C 2**	**C 3**	**Explanation**
Correctness				
Error Handling				
Maintainability				
Runtime Complexity				
Space Complexity				

Answers will vary. However, in general, Candidate 1 was designed to be the most well-documented and maintainable, Candidate 2 was designed to be best at meeting the exact prompt, and Candidate 3 always had a defect. In general, we expect no attributions to Candidate 3, while 1 would be most maintainable, but 2 handles errors the best. Space and Runtime complexity will vary based on implementation you received.

(c) (*3pts.*) Support or refute the claim that you, as the interviewer, must detect and guard against defective code from candidates. Use two sentences or fewer.

Support: You want to reject bad candidates, so you are willing to accept false negatives. The interview is the only time the candidate has to demonstrate competence, so they must provide correct code.

Refute: Increased tension and cognitive load from a non-indicative interview environment may lead the candidate to make small mistake that do not represent their typical behavior. You may be willing to forego a limited amount of correctness if it means evaluating the candidate as a potential co-worker. In addition, you may not be able to guard against incorrect code — can you really interpret every candidate's code perfectly?

6 Fault Localization (20 points)

The following code attempts to determine if, given a list of integers and strings, the strings follow the same pattern as the integers. For example, the input `pattern="1 1 2", string="fish fish dog"` would return *true*, but `pattern="1 2", string="cat cat"` would return *false*.

```
1
2  def wordPattern ( pattern , string ):
3    pattern = pattern.strip ()
4    pattern = pattern.split (' ')
5    string = string.strip ()
6    string = string.split (' ')
7
8    if len ( string ) != len ( pattern ):
9      return False
10
11   match = {}
12
13   for i in range ( len ( pattern )):
14     if pattern[i] in match:
15       if match[ pattern[ i ]] != string [i]:
16         return False
17
18     else :
19       if string [ i ] != match.values() ():
20         match [ pattern [ i ] ] = string [ i ]
21       else :
22         return False
23
24   return True
```

(a) (*15pts.*) Given the five test cases below, identify the *actual* output (produced by the program above) and the *expected* output (based on the problem specification). *Then*, for each test case, specify which lines are *not* covered by that test case. Note the spaces in inputs.

pattern	string	Actual	Expected	Lines *not* covered
`' '`	`'dog dog'`	False	False	
`'1 1 1 1'`	`'dog dog dog dog'`	True	~~False~~ True	
`'1 2 3 2 '`	`''`	False	False	
`'1 2 3 2 '`	`' '`	False	False	
`'1 1 1'`	`'dog cat cat dog'`	False	False	

Lines not covered will vary.

(b) (*2pts.*) Identify any defect(s) in the code above including line number(s) and how to fix it/them. If none are present, explain why.

Answers will vary. We inserted random one-edit mutations.

(c) (*3pts.*) Support or refute the claim that and automated program repair tool could easily apply to the program above with the given fault localization and test suite.

Support: one-edit mutations are readily made by APR tools.

Refute: a poor test suite may mislead fault localization (incorrect suspiciousness), leading to incorrect or inappropriate APR results.

7 Extra Credit **(1 pt each; we are tough on reading questions)**

(a) *(Feedback)* What advice would you give to future students in this class?

u should take it lol

(b) *(Feedback)* What was your least favorite topic covered in the course?

The end.

(c) *(Feedback)* What was your most favorite topic covered in the course?

ALL OF IT.

(d) *(Psychology)* Explain the illusory truth effect in your own words.

repeating a lie a lot makes you believe it's true.

(e) *(Your Choice Reading)* Identify any **optional** reading. Write a sentence about it that convinces us that you read it critically. (Our subjective judgment applies here!).

(f) *(Your Choice Reading 2)* Identify any different **optional** reading. Write a sentence about it that convinces us that you read it critically. (Our subjective judgment applies here!).