CompSci 201, L12: Debugging and Testing (and more Linked List)

Person in CS: Barbara Liskov

- Turing Award Winner in 2008 for contributions to practical and theoretical foundations of programming language and system design, especially related to data abstraction, fault tolerance, and distributed computing.
- "The advice I give people in general is that you should figure out what you like to do, and what you can do well—and the two are not all that dissimilar, because you don't typically like doing something if you don't do it well. ... So you should instead watch—be aware of what you're doing, and what the opportunities are, and step into what seems right, and see where it takes you."



Announcements, Coming up

- Wednesday, 2/22 (today)
 - Project 3: DNA (Linked List) available, due March 6
 - APT Quiz deadline extended to tomorrow Th 2/23.
 - Quiz, no extra grace/late period!
- Monday 2/27
 - Nothing due
- Next Wednesday 3/1
 - APT 5 (linked list problems) due

Ed Discussion Guidelines

Going to review <u>the README</u>

Today's agenda

- 1. Wrapping up linked list problems
 - Reverse in-place
- 2. Testing & Debugging, Concepts & Tools
- 3. More DIYLinkedList

Canonical Linked List Problem

- How do we reverse nodes in a linked list (without creating a new list)?
 - Go from A->B->C to C->B->A
 - Typical interview style question
 - <u>https://leetcode.com/problems/reverse-linked-list/</u>
 - <u>https://www.hackerrank.com/challenges/revers</u>
 <u>e-a-linked-list</u>





Methodical Development

- Turn list = ['A', 'B', 'C'] into
 - rev = ['C', 'B', 'A']
- Move one node at a time, *no new nodes*!
 - Iterative/loop solution with invariant



Invariant to help reason about code

- An invariant is some property that is true each loop guard check (top of the loop)
 - May become false part way through loop
 - Always re-established before guard check
- Example: After k iterations, rev points to the reverse of the first k nodes.
 - before loop iterates at all? rev = null
 - Then at the end we just return rev

one node at a time, assume invariant!

- After 1 iteration: rev is reverse of the first 1 elements
 - list has moved to represent [B,C]
 - So rev represents [A]
- How to move B to front?
- Save a temp reference:
 - Don't lose C-node!



rev = [A], list = [B,C], change: [B,A], [C]

- Pictures and code
 - 1.temp = list.next (so we don't lose C)
 - 2.list.next = rev (make B point to A)
 - 3.rev = list
 - 4.list = temp

(make B point to A) (rev now points to B) (list now points to C)



Working code

```
public ListNode reverse(ListNode front) {
    ListNode rev = null;
    ListNode list = front;
    while (list != null) {
        ListNode temp = list.next;
        list.next = rev;
        rev = list;
        list = temp;
                              rev
    return rev;
                              list →
```

WOTO Go to <u>duke.is/n9pg5</u>

Not graded for correctness, just participation.

Try to answer *without* looking back at slides and notes.

But do talk to your neighbors!



Consider the reverse method. If front is a ListNode with front.next == null, then reverse(front) will... *

10	<pre>public static ListNode reverse(ListNode</pre>
11	ListNode rev = null;
12	<pre>ListNode list = front;</pre>
13	<pre>while (list != null) {</pre>
14	<pre>ListNode temp = list.next;</pre>
15	<pre>list.next = rev;</pre>
16	<pre>rev = list;</pre>
17	<pre>list = temp;</pre>
18	}
19	return rev;
20	}



2

Return null

- Return the same node
- Return a copy of the same node

Same reverse method. We claimed that the following is a loop invariant: "After k iterations, rev points to the reverse of the first k nodes."

For an input list with N elements, which of the following is also a loop invariant for this method? "After k iterations, list..." \ast



Points to the **first k** elements of the input list

3

- Points to the **first N-k** elements of the input list
- Points to the **last k** elements of the input list
- Points to the **last N-k** elements of the input list

4

Same reverse method. Suppose we have a linked list as follows:

list --> 1 --> 2 --> 3 --> null

What will be returned by reverse(<u>list.next</u>)? *



Nothing, null pointer exception occurs

5

Same reverse method. Suppose we have a linked list as follows:

list --> 1 --> 2 --> 3 --> null

In the main program. What will list (also in the main program) be after running reverse(<u>list.next</u>)? *

10	F	ublic static ListNode reverse(ListNo	ode
11		ListNode rev = null;	
12		<pre>ListNode list = front;</pre>	
13		<pre>while (list != null) {</pre>	
14		<pre>ListNode temp = list.next;</pre>	
15		list.next = rev;	
16		<pre>rev = list;</pre>	
17		<pre>list = temp;</pre>	
18		}	
19		return rev;	
20]		
🔵 nu	III		
0 1	> null		
() 1> 2> null			
0 1	○ 1> 2> 3> null		
	Nothing, null pointer exception occurs		

Testing and Debugging

An Algorithmic Problem-Solving Process: UPIC



Not really a linear process



So, something is not correct. Could be...

- The plan (algorithm) did not match the understanding.
- 2. The implementation does not match the plan.
- 3. The understanding was not correct.

First approach to correctness

- Natural temptation to rely on reading source code to verify correctness.
- Like editing an essay for a class, read and check that it makes sense, look for typos.

• But...

```
1 // This is an example of a single line comment using two slashes
 2
3
   /*
    * This is an example of a multiple line comment using the slash and asterisk.
    * This type of comment can be used to hold a lot of information or deactivate
    * code, but it is very important to remember to close the comment.
 6
 7
 8
9
   package fibsandlies:
10
11 import java.util.Map;
12 import java.util.HashMap;
13
14 /**
15
    * This is an example of a Javadoc comment; Javadoc can compile documentation
    * from this text. Javadoc comments must immediately precede the class, method,
    * or field being documented.
17
18
    * @author Wikipedia Volunteers
19
    */
20
   public class FibCalculator extends Fibonacci implements Calculator {
21
       private static Map<Integer, Integer> memoized = new HashMap<>();
22
23
        /*
24
        * The main method written as follows is used by the JVM as a starting point
25
        * for the program.
26
        */
27
       public static void main(String[] args) {
28
            memoized.put(1, 1);
29
            memoized.put(2, 1);
30
            System.out.println(fibonacci(12)); // Get the 12th Fibonacci number and print to console
31
       }
32
33
34
        * An example of a method written in Java, wrapped in a class.
35
        * Given a non-negative number FIBINDEX, returns
36
        * the Nth Fibonacci number, where N equals FIBINDEX.
37
38
        * @param fibIndex The index of the Fibonacci number
39
        * @return the Fibonacci number
40
41
        public static int fibonacci(int fibIndex) {
42
            if (memoized.containsKey(fibIndex)) {
43
                return memoized.get(fibIndex);
44
           }
45
46
            int answer = fibonacci(fibIndex - 1) + fibonacci(fibIndex - 2);
47
            memoized.put(fibIndex. answer):
48
            return answer:
49
       }
50 }
```

Code is complex and interrelated

Miss something in your essay? The rest of the essay may still make sense?

One thing wrong in the code? Could prevent the whole program from functioning. And code gets complicated!



A tale of two programmers...

Too confident

"I'm amazing at programming, I don't need to test my code because I **know** it's correct."

The beginning of a security vulnerability, broken app, ...

Low confidence

"My code doesn't work, that must be because I'm personally bad at this. There is no way I could figure this out myself."

> Mistaken expectations, Feeling helpless, not sure what to do

What is testing?

Verifying that an implementation *functions* as *expected*.

- What is functionality is expected?
- Given an input, what output is expected?

Can test at multiple levels: single method (*unit*), class (*integration*), whole project (*integration/functionality*), ...

Black box testing (can run program, can't see source code) and *white box testing* (access to source code).

SandwichBar APT Example

Given:

- String[] available, a list of ingredients the sandwich bar can use, and
- String[] orders, the types of sandwiches I like, in order of preference (most preferred first)

return the O-based index of the sandwich I will buy. If the bar can make no sandwiches I like, return -1.

Example:

- available: { "ham", "cheese", "mustard" }
- orders: { "ham cheese" }
- Should return: 0

The first test: the compiler

- Compiler performs *static* analysis; check for errors detectable in the source code *before running*.
 - Often *type errors* (e.g., trying to assign a String to an int, trying to treat an Array as a list, ...)

6 7 8 9	<pre>public int whichOrder(String[] available, String[] orders){ for (int i=0; i<orders.length; (canmake(available,="" i++)="" if="" orders[i]))="" orders[i];="" return="" th="" {="" }="" }<=""></orders.length;></pre>			
PRO	BLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL			
~ (SandwichBar.java 1			
	\otimes Type mismatch: cannot convert from String to int Java(16777235) [9, 24]			

Manual test

- Given an input, what is the expected output?
- Run program with expected input. What do you get?

```
Run | Debug
 25
        public static void main(String[] args) {
            String[] testAvailable = { "ham", "cheese", "mustard" };
 26
 27
            String[] testOrders = { "ham cheese" };
 28
            SandwichBar testInstance = new SandwichBar();
 29
            int testResult = testInstance.whichOrder(testAvailable, testOrders);
 30
            System.out.println(testResult);
 31
        }
 32
                                                     I expect the code to return 0,
 22
                                                     example from before. And it
PROBLEMS
              OUTPUT
                      DEBUG CONSOLE
                                      TERMINAL
                                                    does! My solution must work!
 0
```

How many tests are enough?

	Run l	Debug			
25	<pre>public static void main(String[] args) {</pre>				
26		<pre>String[] testAvailable = { "cheese", "mustard", "lettuce" };</pre>			
27		<pre>String[] testOrders = { "cheese ham", "cheese mustard lettuce", "ketchup", "beer" };</pre>			
28		<pre>SandwichBar testInstance = new SandwichBar();</pre>			
29		<pre>int testResult = testInstance.whichOrder(testAvailable, testOrders);</pre>			
30		<pre>System.out.println(testResult);</pre>			
31	}	-			
ROBLE	MS 2	OUTPUT	DEBUG CONSC	That's not right, I can't make a ham	xt, !exclude)
0				and choose sandwich without ham	

- Can never have enough tests to guarantee correctness, but...
- More and more diverse tests can help increase confidence.



Automated testing?

For when you want to run many tests without doing it manually one at a time...automate it!

Ways you *use* automated testing in 201:

- Junit tests Junit is a popular external library, no built-in standard library unit testing in Java.
- Gradescope autograder
- APT server

In professional software development? You also write the tests!

Test early, test small, test often

- Unit testing: Term for tests conducted on the smallest *units* of code that take inputs and produce outputs.
 - In Java, typically methods, preferably short ones (10-20 lines). Test as soon as you write, don't wait!
 - Method getting too complex? Helper method!



Debugging



Debugging loop:

- Detect unexpected behavior through testing.
- 2. Isolate *cause* of unexpected behavior.
- 3. Change implementation.
- 4. Test again.

How to isolate the cause of unexpected behavior

- Want to identify the *first point of divergence from expected behavior*.
 - May have started long before your test result!
- Try to answer the question:
 - What is the *first* line of code in which method of which class that first did something different than I expected?
 - Never fixate on line 30 if you're not sure lines 1-29 are working.

Debugging Methods

- Three common methods:
 - Examine code and small examples by hand
 - Add print statements to code
 - Use a debugger tool

To step through execution line by line Good start, might get complicated

Allows you to see the *state* of the program while running. Tip: Can add print statements for APTs, will show up on server!

• We have already seen the basic debugger tool built into an extension on your visual studio code. Will review in detail today.

Debugger tool



- Instead of run? Choose debug!
- Walk through execution of program *line by line*.
- See current state of all variables line by line.

Set a breakpoint



- Start by setting a *breakpoint* in your code.
- Says "run the program until the first time this line executes, then pause to step line by line."
- If you want to go line by line from the beginning? Set to first line in main.

Debug options

Will see a menu like this:

- Continue: Go to next breakpoint
- Step over: Execute line, go to next. Run whole methods.
- Step into: Same as over unless method call. Steps into methods, jumping to first line of method code.
 - Step out: Break out of method back to where called
 - Restart: Start over again at first breakpoint
 - Stop: Stop debugging session

State of program

5 public class SandwichBar {

D

6

7 8 if (canMake(available, orders[i])) {

\vee VARIABLES

\sim Local

```
✓ available: String[3]@15
```

- > 0: "cheese"
- > 1: "mustard"
- > 2: "lettuce"
- ✓ orders: String[4]@16
 - > 0: "cheese ham"
 - > 1: "cheese mustard lettuce"
 - > 2: "ketchup"
 - > 3: "beer"

Can see all values of all local variables while executing at highlighted line.

Can step through to determine *first* time values diverge from expectations.

Debugging linked list?

\sim VARI/ \sim Loc	ARIABLES ARIABLES ARIAB	ested "list"		
ar	args: String[0]@8 of object refer	ences.		
> my	> myNums: int[3]@10			
∨ my	✓ myNumsList: ListNode@12			
i	• Expand one no	nde at a		
\sim r	v next: ListNode@15	ac at a		
	info: 0 time.			
\sim	✓ next: ListNode@16			
	info: 1			
	next: null			
46	<pre>public static void main(String[] args) { args =</pre>	String[0]@8		
47	47 int[] myNums = {2, 0, 1}; myNums = int[3]@10			
48	<pre>ListNode myNumsList = listFromArray(myNums);</pre>	myNumsList =		
49	<pre>printList(myNumsList); myNumsList = ListNode</pre>	<mark>@12</mark>		
50	<pre>Ø System.out.println(getVal(myNumsList, 1));</pre>			
51	}			

Want something more visual?

pythontutor.com/java.html



Can use if you need to visualize stepping through some pointer code.

Test Results Follow (scro

of correct: 30 out of 30

1	pass	
2	pass	
3	pass	
4	pass	
5	pass	
6	pass	
7	pass	
8	pass	
9	pass	
10	pass	
11	pass	
12	pass	
13	pass	
14	pass	
15	pass	
16	pass	
17	pass	
18	pass	
19	pass	
20	Doca	

Debugging reflection

Goal is to become a more *active* and *empowered* tester and debugger.

- Build confidence as you develop.
- Take *active* steps to isolate the problem
- Test, use the debugger, gather data, reason about it
- Less time staring at the code, feeling frustrated

More DIYLinkedList





- Writing unit tests?
- Add to arbitrary index?
- Efficient iterator?