

**Solution (Instructor only):***Solution for Graph II:*(a) See the graph tool at <http://www.cs.gmu.edu/~offutt/softwaretest/>(b)  $x$  has 6 du-paths, as enumerated below:

|            |                 |
|------------|-----------------|
| <i>i</i>   | [1, 2, 3]       |
| <i>ii</i>  | [1, 2, 6]       |
| <i>iii</i> | [3, 4, 5, 2, 3] |
| <i>iv</i>  | [3, 4, 5, 2, 6] |
| <i>v</i>   | [3, 5, 2, 3]    |
| <i>vi</i>  | [3, 5, 2, 6]    |

(c) The numbers in the table below correspond to the du-paths in the previous table. The table indicates whether each test path tours each du-path with or without a sidetrip.

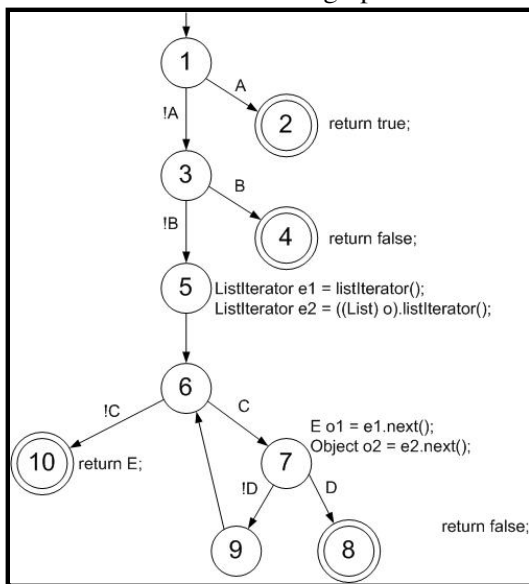
|       | <i>direct</i>     | <i>w/ sidetrip</i> |
|-------|-------------------|--------------------|
| $t_1$ | <i>ii</i>         |                    |
| $t_2$ | <i>i, iii, vi</i> |                    |
| $t_3$ | <i>i, iv, v</i>   |                    |
| $t_4$ | <i>i, vi</i>      |                    |

Note that, for example, although  $t_2$  does not tour path (v) with a sidetrip, it does tour path (v) with a detour.

- (d) This question has multiple possible answers. All test paths use the def in 1, and test paths  $\{t_2\}$ ,  $\{t_3\}$ ,  $\{t_4\}$  each use the def in 3. Possible answers:  $\{t_2\}$ ,  $\{t_3\}$ , or  $\{t_4\}$ .
- (e) This question only has two possible answers.  $\{t_1\}$  is required for the def in 1 to reach the use in 6. Either  $t_2$  or  $t_3$  is required for the def in 3 to reach the use in 3. Possible answers:  $\{t_1, t_2\}$  or  $\{t_1, t_3\}$ .
- (f) This question has one possible answer:  $\{t_1, t_2, t_3\}$ .  $t_1$  is required for path (ii).  $t_2$  is required for path (iii).  $t_3$  is required for path (iv). Since  $t_1$ ,  $t_2$ , and  $t_3$  together tour all six du-paths,  $t_4$  is not needed.

1.

**Answer:** Here is a 10 node graph.



2.

```

A: o == this
B: !(o instanceof List)
C: e1.hasNext() && e2.hasNext()
D: !(o1==null ? o2==null : o1.equals(o2))
E: !(e1.hasNext() || e2.hasNext())
  
```

**Answer:** See prior question.

3.

**Answer:** There are 4 separate return statements. Reaching each one requires a test case.

4.

**Answer:**

```
// Test 1: (Expected==true) Reaches first return statement
list1.equals(list1);

// Test 2: (Expected==false) Reaches second return statement
list1.equals(null);

// Test 3: (Expected==true) Reaches fourth return statement
// Note that nodes inside loop are also visited
list1.add("ant");
list2.add("ant");
list1.equals(list2);

// Test 4: (Expected==false) Reaches third return statement
list1.add("ant");
list2.add("bat");
list1.equals(list2);
```