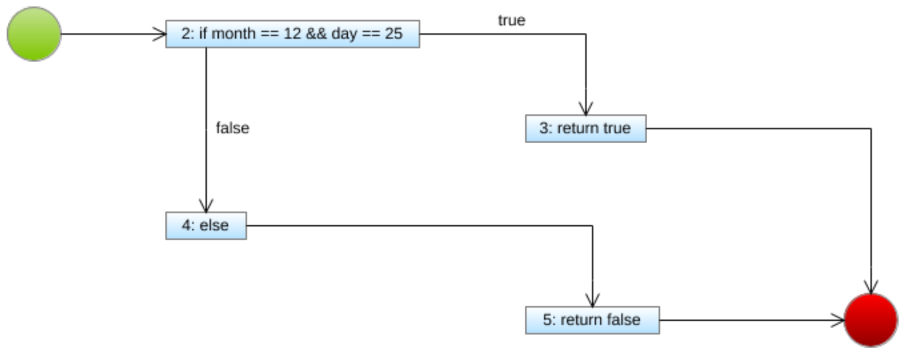
Solutions - Is It Xmas?

Answer the following questions for each of the above example methods and test suites.

Control-Flow Graph: Draw a control-flow graph for the function. In addition to the usual CFG features, label each node with the corresponding code-line number.



Statement Coverage: For each test in the test suite, list the nodes covered by the test case.

Test 1) 2, 3

Test 2) 2, 4, 5

Test 3) ____________________

Test 4) ____________________

Test 5) ____________________

Test 6) ____________________

Does the test suite achieve **statement coverage**? **Yes** No

If "No", which nodes did the test suite miss? ____________________

Branch Coverage: For each test in the test suite, list the relevant edges covered by the test case. Denote an edge like this, 2→3, which denotes the edge from node 2 to node 3.

Test 1) 2→3

Test 2) 2→4

Test 3) ____________________

Test 4) ____________________

Test 5) ____________________

Test 6) ____________________

Does the test suite achieve **branch coverage**? **Yes** No

If "No", which edges did the test suite miss? ____________________

Path Coverage: First, list all the paths through the CFG. For each path, list the sequence of nodes on the path, like this: 2→3→5→6, which denotes a path from node 2 to node 6. You need only cover executions that involve at most 1 iteration of each loop (if there are any loops). There may be more lines below than there are paths; use only as many lines as you need.

2→3

2→4→5

For each test in the test suite, list the path covered by the test case.

Test 1) 2→3

Test 2) 2→4→5

Test 3)

Test 4)

Test 5)

Test 6)

Does the test suite achieve **path coverage**? **Yes** No

If "No", which paths did the test suite miss? ______

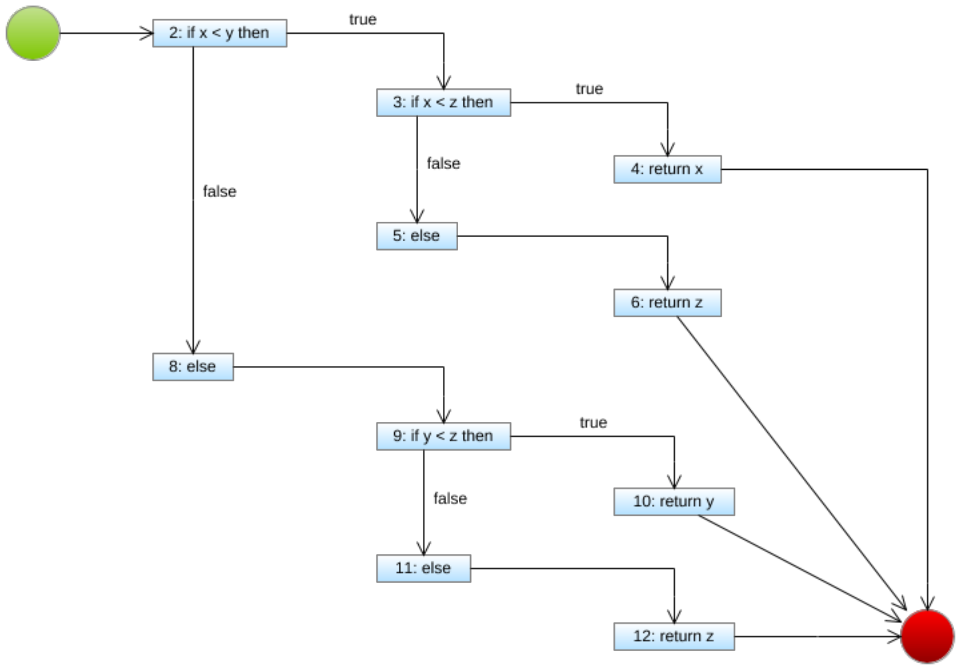
Would the example test suite detect the bug in the buggy variant of the code? If so, which test(s) would fail?

No, the example test suite would not detect the bug

Solutions - Min of Three

Answer the following questions for each of the above example methods and test suites.

Control-Flow Graph: Draw a control-flow graph for the function. In addition to the usual CFG features, label each node with the corresponding code-line number.



Statement Coverage: For each test in the test suite, list the nodes covered by the test case.

Test 1) 2, 3, 4

Test 2) 2, 3, 5, 6

Test 3) 2, 8, 9, 10

Test 4) 2, 8, 9, 11, 12

Test 5) ______

Test 6) ______

Does the test suite achieve **statement coverage**? **Yes** No

If "No", which nodes did the test suite miss? ______

Branch Coverage: For each test in the test suite, list the relevant edges covered by the test case. Denote an edge like this, 2→3, which denotes the edge from node 2 to node 3.

Test 1) 2→3, 3→4

Test 2) 2→3, 3→5

Test 3) 2→8, 9→10

Test 4) 2→8, 9→11

Test 5) ______

Test 6) ______

Does the test suite achieve **branch coverage**? **Yes** No

If "No", which edges did the test suite miss? ______

Path Coverage: First, list all the paths through the CFG. For each path, list the sequence of nodes on the path, like this: 2→3→5→6, which denotes a path from node 2 to node 6. You need only cover executions that involve at most 1 iteration of each loop (if there are any loops). There may be more lines below than there are paths; use only as many lines as you need.

2→3→4

2→3→5→6

2→8→9→10

2→8→9→11→12

For each test in the test suite, list the path covered by the test case.

Test 1) 2→3→4

Test 2) 2→3→5→6

Test 3) 2→8→9→10

Test 4) 2→8→9→11→12

Test 5)

Test 6)

Does the test suite achieve **path coverage**? **Yes** No

If "No", which paths did the test suite miss? ______________________

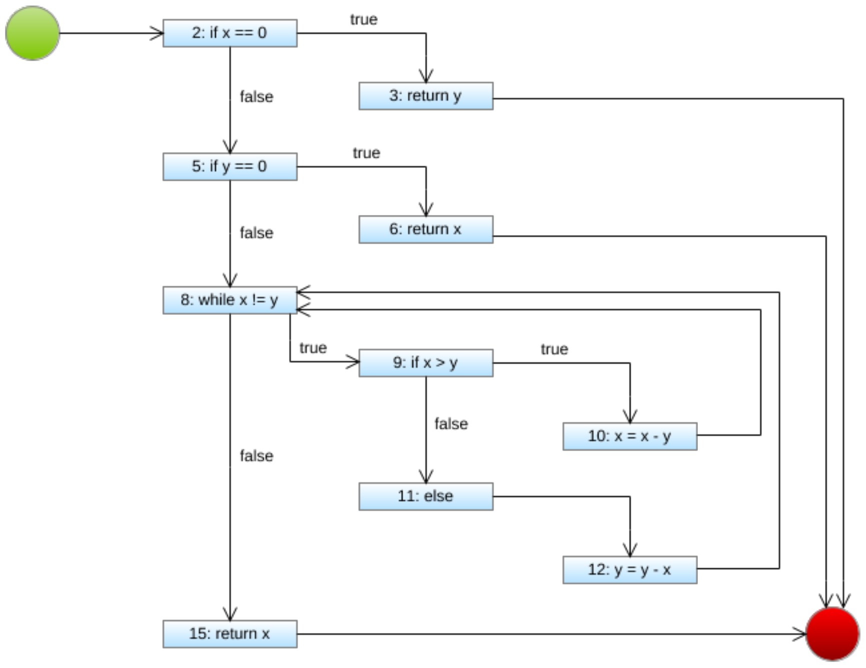
Would the example test suite detect the bug in the buggy variant of the code? If so, which test(s) would fail?

Yes, test #3 would fail (returns 2 when expected is 1)

Solutions - Greatest Common Divisor

Answer the following questions for each of the above example methods and test suites.

Control-Flow Graph: Draw a control-flow graph for the function. In addition to the usual CFG features, label each node with the corresponding code-line number.



Statement Coverage: For each test in the test suite, list the nodes covered by the test case.

Test 1) 2, 3

Test 2) 2, 5, 6

Test 3) 2, 5, 8, 15

Test 4)

Test 5)

Test 6)

Does the test suite achieve **statement coverage**? **Yes** No

If "No", which nodes did the test suite miss? ______________________

Branch Coverage: For each test in the test suite, list the relevant edges covered by the test case. Denote an edge like this, 2→3, which denotes the edge from node 2 to node 3.

Test 1) 2→3

Test 2) 2→5, 5→6

Test 3) 2→5, 5→8, 8→15

Test 4) 2→5, 5→8, 8→9, 9→10, 8→15

Test 5) 2→5, 5→8, 8→9, 9→11, 8→15

Test 6)

Does the test suite achieve **branch coverage**? **Yes** No

If "No", which edges did the test suite miss? ______________________

Path Coverage: First, list all the paths through the CFG. For each path, list the sequence of nodes on the path, like this: 2→3→5→6, which denotes a path from node 2 to node 6. You need only cover executions that involve at most 1 iteration of each loop (if there are any loops). There may be more lines below than there are paths; use only as many lines as you need.

2→3

2→5→6

2→5→8→15

2→5→8→9→10→8→15

2→5→8→9→11→12→8→15

For each test in the test suite, list the path covered by the test case.

Test 1) 2→3

Test 2) 2→5→6

Test 3) 2→5→8→15

Test 4) 2→5→8→9→10→8→15

Test 5) 2→5→8→9→11→12→8→15

Test 6)

Does the test suite achieve **path coverage**? **Yes** No

If "No", which paths did the test suite miss? ______

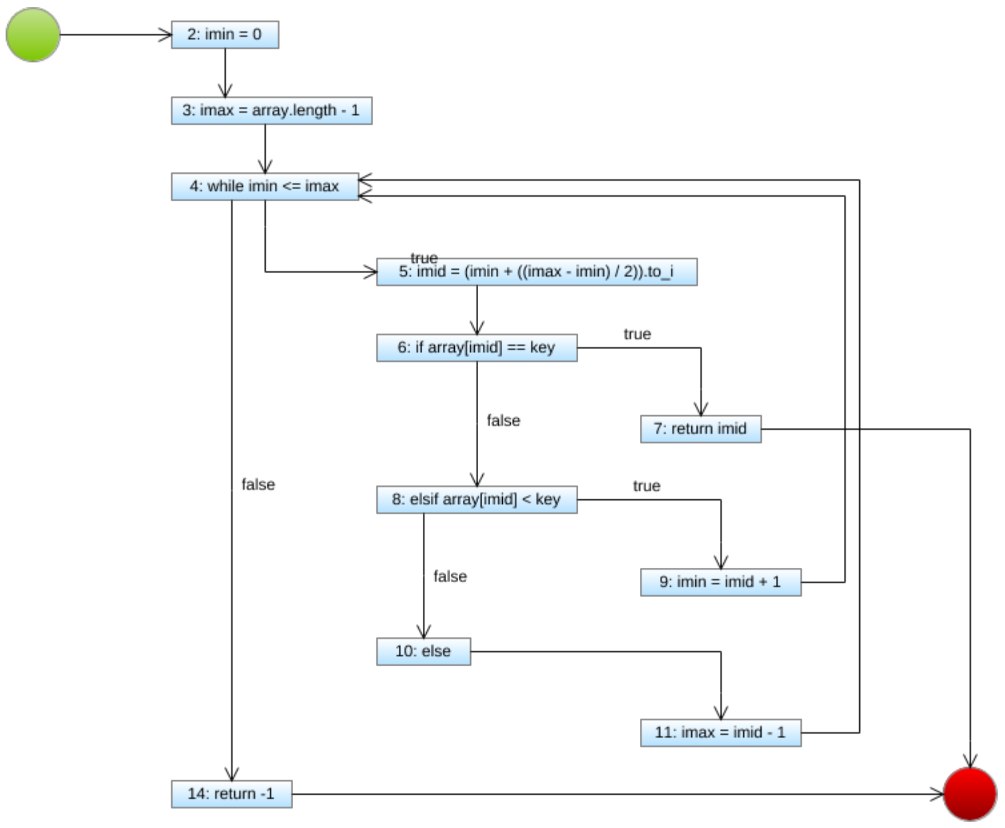
Would the example test suite detect the bug in the buggy variant of the code? If so, which test(s) would fail?

No, the example test suite would not detect the bug

Solutions - Binary Search

Answer the following questions for each of the above example methods and test suites.

Control-Flow Graph: Draw a control-flow graph for the function. In addition to the usual CFG features, label each node with the corresponding code-line number.



Statement Coverage: For each test in the test suite, list the nodes covered by the test case.

Test 1) 2, 3, 4, 14

Test 2) 2, 3, 4, 5, 6, 8, 9, 14

Test 3) 2,3 4, 5, 6, 8, 10, 11, 14

Test 4) 2, 3, 4, 5, 6, 7

Test 5) ______

Test 6) ______

Does the test suite achieve **statement coverage**? **Yes** No

If "No", which nodes did the test suite miss? ______

Branch Coverage: For each test in the test suite, list the relevant edges covered by the test case. Denote an edge like this, 2→3, which denotes the edge from node 2 to node 3.

Test 1) 4→14

Test 2) 4→5, 6→8, 8→9, 4→14

Test 3) 4→5, 6→8, 8→10, 4→14

Test 4) 4→5, 6→7

Test 5) ______

Test 6) ______

Does the test suite achieve **branch coverage**? **Yes** No

If "No", which edges did the test suite miss? ______

Path Coverage: First, list all the paths through the CFG. For each path, list the sequence of nodes on the path, like this: 2→3→5→6, which denotes a path from node 2 to node 6. You need only cover executions that involve at most 1 iteration of each loop (if there are any loops). There may be more lines below than there are paths; use only as many lines as you need.

2→3→4→14

2→3→4→5→6→7

2→3→4→5→6→8→9→4→14

2→3→4→5→6→8→10→11→4→14

For each test in the test suite, list the path covered by the test case.

Test 1) 2→3→4→14

Test 2) 2→3→4→5→6→8→9→4→14

Test 3) 2→3→4→5→6→8→10→11→4→14

Test 4) 2→3→4→5→6→7

Test 5)

Test 6)

Does the test suite achieve **path coverage**? **Yes** No

If "No", which paths did the test suite miss? ______

Would the example test suite detect the bug in the buggy variant of the code? If so, which test(s) would fail?

No, the example test suite would not detect the bug