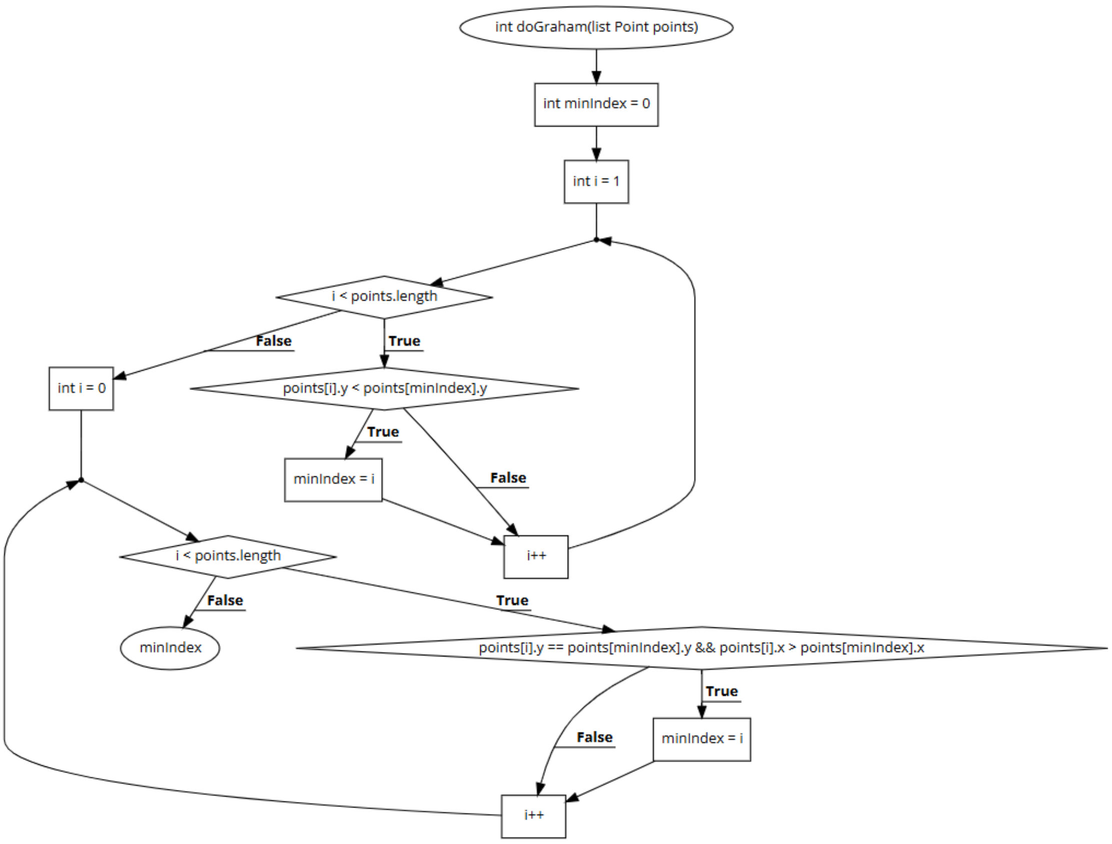
Software Engineering (IT-314)

ID: 202201238

Name: Riddhi Mistry

Lab – 9

Q-1] Convert the code comprising the beginning of the doGraham method into a control flow graph (CFG). You are free to write the code in any programming language.



Q-2] Construct test sets for your flow graph that are adequate for the following criteria:

Statement Coverage

Statement Coverage requires that each statement in the code is executed at least once.

Input: points = [{x: 1, y: 1}, {x: 2, y: 0}, {x: 3, y: 2}]

Expected Output: 1

In this test case loop-1 is executed.

input: points = [{x: 2, y: 0}, {x: 3, y: 0}]

Expected Output: 1

In this test case loop-2 is executed.

Branch Coverage

Branch Coverage requires that every branch (i.e., each if condition) evaluates to both true and false at least once.

Input: points = [{x: 0, y: 0}, {x: 2, y: 2}, {x: 3, y: 1}]

Expected Output: 0

In this test case first if condition true.

Input: points = [{x: 2, y: 1}, {x: 4, y: 1}, {x: 1, y: 1}]

Expected Output: 1

In this test case second if condition true.

Basic Condition Coverage

Basic Condition Coverage requires that each condition in a decision is tested with both true and false outcomes independently.

Input: points = [{x: 1, y: 2}, {x: 3, y: 0}, {x: 5, y: 2}]

Expected Output: 1

the condition points[i].y < points[minIndex].y will be true when i = 1 and false for other iterations.

Input: points = [{x: 2, y: 1}, {x: 5, y: 1}, {x: 4, y: 1}]

Expected Output: 1

The condition points[i].y == points[minIndex].y && points[i].x > points[minIndex].x will be true for points[1] and points[2] but false for the initial point, testing all conditions in the second loop.

Q-3] And Q -4]

```
!pip install mutpy
```

```
import unittest

class Point:
    def __init__(self, x, y):
        self.x = x
        self.y = y

def do_graham(points):
    minindex = 0

    for i in range(1, len(points)):
        if points[i].y < points[minindex].y:
            minindex = i

    for i in range(len(points)):
        if points[i].y == points[minindex].y and points[i].x >
points[minindex].x:
            minindex = i

    return minindex

class TestGrahamFunction(unittest.TestCase):

    def test_single_point(self):
        points = [Point(1, 2)]
        result = do_graham(points)
        self.assertEqual(result, 0)

    def test_multiple_points(self):
        points = [Point(2, 3), Point(1, 1), Point(3, 1), Point(0, 2)]
        result = do_graham(points)
        self.assertEqual(result, 2)

    def test_same_y_coordinate(self):
        points = [Point(1, 1), Point(2, 1), Point(3, 1)]
        result = do_graham(points)
        self.assertEqual(result, 2)

    def test_negative_coordinates(self):
        points = [Point(-1, -2), Point(-3, -1), Point(-2, -3)]
        result = do_graham(points)
        self.assertEqual(result, 2)

    def test_mixed_coordinates(self):
        points = [Point(1, 2), Point(-1, -1), Point(0, 0), Point(2, -
2)]
        result = do_graham(points)
        self.assertEqual(result, 3)

# Run the tests explicitly
if __name__ == '__main__':
    suite =
unittest.TestLoader().loadTestsFromTestCase(TestGrahamFunction)
    runner = unittest.TextTestRunner()
    runner.run(suite)
```

output:

```
.....
----------------------------------------------------------------------
Ran 5 tests in 0.020s

OK
```

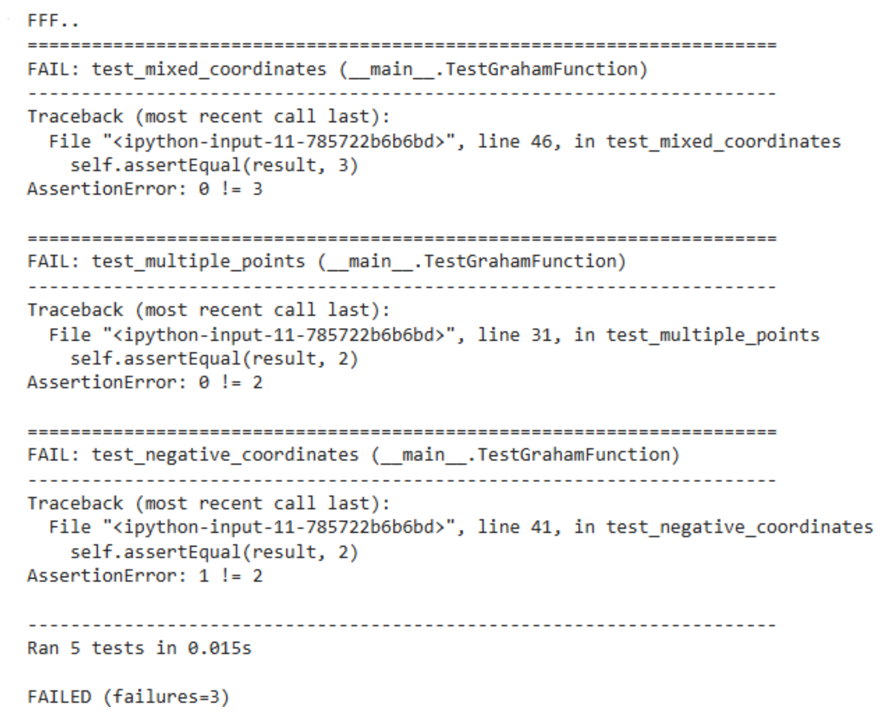
Mut – 1

```
    for i in range(len(points)):
        if points[i].y == points[minindex].y and points[i].x <
points[minindex].x:
            minindex = i
```

```
.F.F.
======================================================================
FAIL: test_multiple_points (__main__.TestGrahamFunction)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "<ipython-input-10-ed3669ad05c4>", line 31, in test_multiple_points
    self.assertEqual(result, 2)
AssertionError: 1 != 2

======================================================================
FAIL: test_same_y_coordinate (__main__.TestGrahamFunction)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "<ipython-input-10-ed3669ad05c4>", line 36, in test_same_y_coordinate
    self.assertEqual(result, 2)
AssertionError: 0 != 2

----------------------------------------------------------------------
Ran 5 tests in 0.012s

FAILED (failures=2)
```

Mut – 2

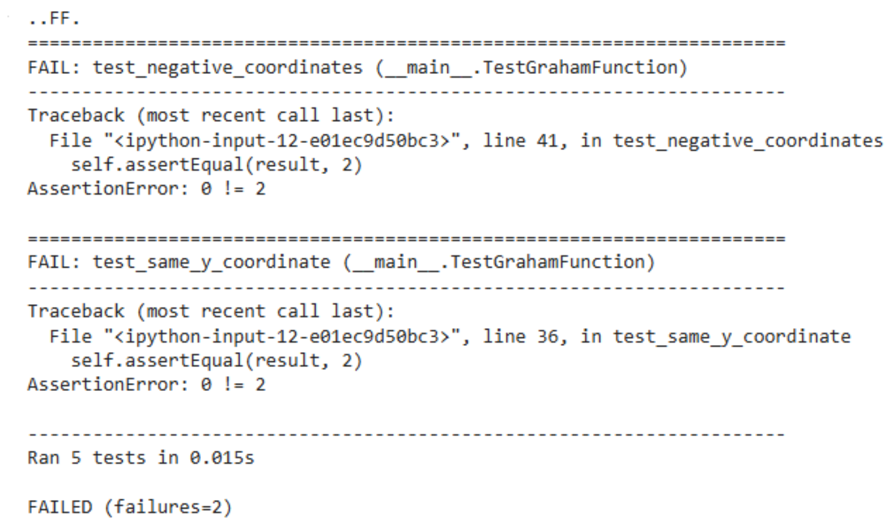
```
    for i in range(1, len(points)):
        if points[i].y > points[minindex].y:
            minindex = i
```



```
FFF..
======================================================================
FAIL: test_mixed_coordinates (__main__.TestGrahamFunction)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "<ipython-input-11-785722b6b6bd>", line 46, in test_mixed_coordinates
    self.assertEqual(result, 3)
AssertionError: 0 != 3

======================================================================
FAIL: test_multiple_points (__main__.TestGrahamFunction)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "<ipython-input-11-785722b6b6bd>", line 31, in test_multiple_points
    self.assertEqual(result, 2)
AssertionError: 0 != 2

======================================================================
FAIL: test_negative_coordinates (__main__.TestGrahamFunction)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "<ipython-input-11-785722b6b6bd>", line 41, in test_negative_coordinates
    self.assertEqual(result, 2)
AssertionError: 1 != 2

----------------------------------------------------------------------
Ran 5 tests in 0.015s

FAILED (failures=3)
```

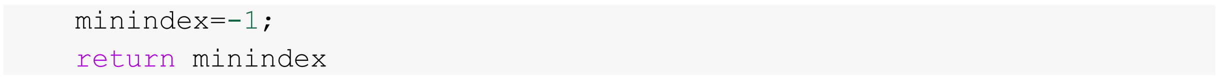
Mut – 3

```
    for i in range(len(points)):
        if points[i].y != points[minindex].y and points[i].x >
points[minindex].x:
            minindex = i
```

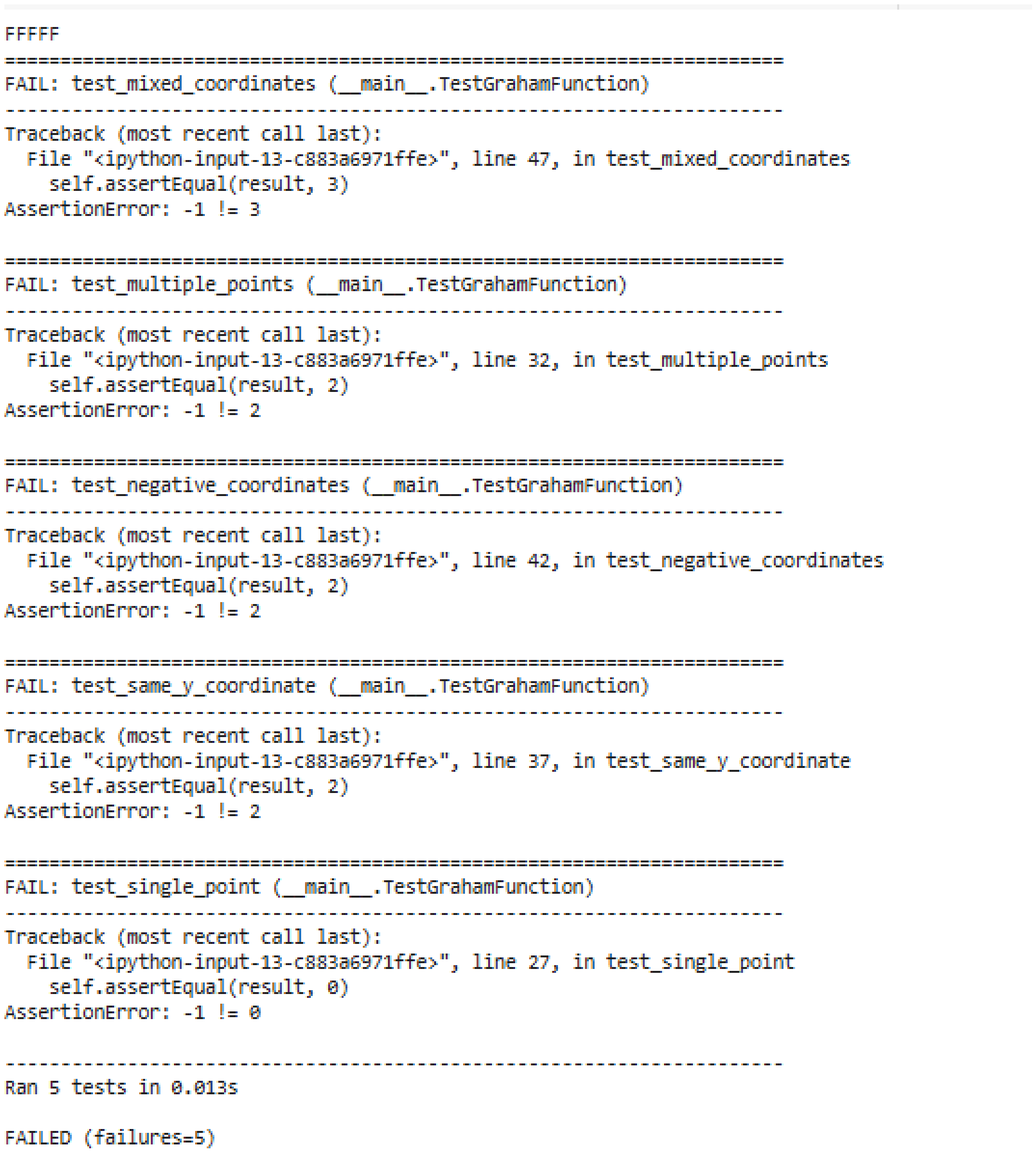


```
..FF.
======================================================================
FAIL: test_negative_coordinates (__main__.TestGrahamFunction)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "<ipython-input-12-e01ec9d50bc3>", line 41, in test_negative_coordinates
    self.assertEqual(result, 2)
AssertionError: 0 != 2

======================================================================
FAIL: test_same_y_coordinate (__main__.TestGrahamFunction)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "<ipython-input-12-e01ec9d50bc3>", line 36, in test_same_y_coordinate
    self.assertEqual(result, 2)
AssertionError: 0 != 2

----------------------------------------------------------------------
Ran 5 tests in 0.015s

FAILED (failures=2)
```

Mut – 4



```
    minindex=-1;
    return minindex
```



```
FFFFF
======================================================================
FAIL: test_mixed_coordinates (__main__.TestGrahamFunction)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "<ipython-input-13-c883a6971ffe>", line 47, in test_mixed_coordinates
    self.assertEqual(result, 3)
AssertionError: -1 != 3

======================================================================
FAIL: test_multiple_points (__main__.TestGrahamFunction)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "<ipython-input-13-c883a6971ffe>", line 32, in test_multiple_points
    self.assertEqual(result, 2)
AssertionError: -1 != 2

======================================================================
FAIL: test_negative_coordinates (__main__.TestGrahamFunction)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "<ipython-input-13-c883a6971ffe>", line 42, in test_negative_coordinates
    self.assertEqual(result, 2)
AssertionError: -1 != 2

======================================================================
FAIL: test_same_y_coordinate (__main__.TestGrahamFunction)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "<ipython-input-13-c883a6971ffe>", line 37, in test_same_y_coordinate
    self.assertEqual(result, 2)
AssertionError: -1 != 2

======================================================================
FAIL: test_single_point (__main__.TestGrahamFunction)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "<ipython-input-13-c883a6971ffe>", line 27, in test_single_point
    self.assertEqual(result, 0)
AssertionError: -1 != 0

----------------------------------------------------------------------
Ran 5 tests in 0.013s

FAILED (failures=5)
```