



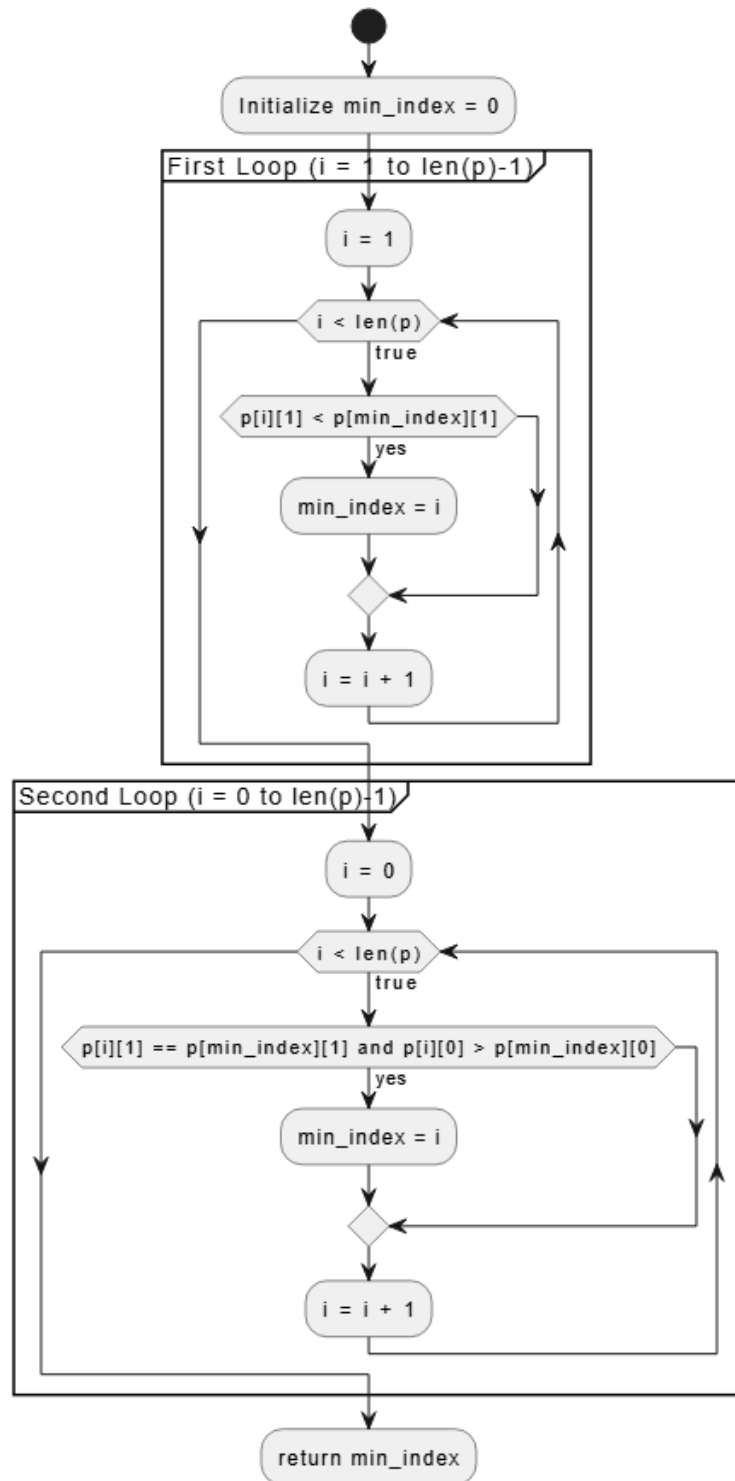
**IT314 – Software Engineering Lab-09**  
**Mutation Testing**  
**Ayush Popshetwar – 202201412**

**Question 1) doGraham Function ()**

**On converting code to python**

```
def do_graham(p):  
    min_index = 0  
  
    # Search for minimum point  
    for i in range(1, len(p)):  
        if p[i][1] < p[min_index][1]: # Compare y values  
            min_index = i  
  
    # Continue along the values with the same y component  
    for i in range(len(p)):  
        if p[i][1] == p[min_index][1] and p[i][0] >  
p[min_index][0]: # Compare x values  
            min_index = i  
  
    return min_index # Return the index of the minimum point
```

**i. Control Flow Graph (CFG):** Yes, the logic of CFG generated matches with the one by tool using Eclipse flow graph generator



## ii. Test Sets

### a. Statement Coverage

To achieve statement coverage, we need to ensure that each statement is executed at least once.

#### Minimum Test Cases:

1. **Test Case 1:**  $p = [(1, 2), (2, 3), (0, 1)]$ 
  - Minimum point is  $(0, 1)$ , index 2.
2. **Test Case 2:**  $p = [(1, 2), (2, 1), (0, 1)]$ 
  - Minimum point is  $(0, 1)$ , index 2.
3. **Test Case 3:**  $p = [(1, 2), (1, 2), (0, 1)]$ 
  - Minimum point is  $(0, 1)$ , index 2.

### b. Branch Coverage

To achieve branch coverage, we need to ensure that both true and false outcomes for each condition are tested.

#### Minimum Test Cases:

1. **Test Case 1:**  $p = [(1, 2), (2, 3), (0, 1)]$  (Covers first condition True)
2. **Test Case 2:**  $p = [(1, 2), (1, 3), (0, 1)]$  (Covers first condition False)
3. **Test Case 3:**  $p = [(1, 1), (2, 1), (0, 1)]$  (Covers second condition True)
4. **Test Case 4:**  $p = [(1, 1), (0, 1), (0, 2)]$  (Covers second condition False)

### c. Basic Condition Coverage

This requires that each atomic condition in the program has been evaluated to be both true and false.

#### Minimum Test Cases:

1. **Test Case 1:**  $p = [(1, 2), (2, 3), (0, 1)]$  (First condition True)
2. **Test Case 2:**  $p = [(1, 2), (1, 3), (0, 1)]$  (First condition False)
3. **Test Case 3:**  $p = [(1, 1), (2, 1), (0, 1)]$  (Second condition True for both)
4. **Test Case 4:**  $p = [(1, 1), (0, 1), (0, 2)]$  (Second condition False for both)

To achieve basic condition coverage, we can reuse the four cases from branch coverage since they effectively cover all atomic conditions.

### iii. Mutation Testing

```
C:\Users\Admin>pip install MutPy==0.3.0
Collecting MutPy==0.3.0
  Downloading MutPy-0.3.0.tar.gz (14 kB)
    Preparing metadata (setup.py) ... done
Collecting PyYAML>=3.1
  Downloading PyYAML-6.0.2-cp311-cp311-win_amd64.whl (161 kB)
----- 162.0/162.0 kB 966.5 kB/s eta 0:00:00
Installing collected packages: PyYAML, MutPy
  DEPRECATION: MutPy is being installed using the legacy 'setup.py install' method, because it does not have a 'pyproject.toml' and the 'wheel' package is not installed. pip 23.1 will enforce this behaviour change. A possible replacement is to enable the '--use-pep517' option. Discussion can be found at https://github.com/pypa/pip/issues/8559
  Running setup.py install for MutPy ... done
Successfully installed MutPy-0.3.0 PyYAML-6.0.2
```

```
C:\Users\Admin\Documents\project>python test_convex_hull.py
.F..
=====
FAIL: test_y_value_different (__main__.TestDoGraham.test_y_value_different)
-----
Traceback (most recent call last):
  File "C:\Users\Admin\Documents\project\test_convex_hull.py", line 12, in test_y_value_different
    self.assertEqual(do_graham(p), 1)
AssertionError: 2 != 1

-----
Ran 4 tests in 0.002s

FAILED (failures=1)
```

Corrected Code:

convex\_hull.py - C:/Users/Admin/Documents/convex\_hull.py (3.10.5)

File Edit Format Run Options Window Help

```
def do_graham(p):
    min_index = 0

    for i in range(len(p)):

        if p[i][1] < p[min_index][1]:
            min_index = i

        elif p[i][1] == p[min_index][1] and p[i][0] < p[min_index][0]:
            min_index = i

    return min_index
```

## Test cases:

test\_convex\_hull.py - C:/Users/Admin/Documents/project/test\_convex\_hull.py (3.1

File Edit Format Run Options Window Help

```
import unittest
from convex_hull import do_graham

class TestDoGraham(unittest.TestCase):
    def test_basic_case(self):
        p = [[0, 0], [1, 1]]
        self.assertEqual(do_graham(p), 0)

    def test_y_value_different(self):
        p = [[0, 0], [1, -1]]
        self.assertEqual(do_graham(p), 1)

    def test_y_value_same_x_different(self):
        p = [[0, 0], [0, 0], [1, 0]]
        self.assertEqual(do_graham(p), 2)

    def test_y_value_edge_case(self):
        p = [[1, 1], [1, 1], [2, 0]]
        self.assertEqual(do_graham(p), 2)

if __name__ == '__main__':
    unittest.main()
```

```
C:\Users\Admin\Documents\project>python test_convex_hull.py
....
-----
Ran 4 tests in 0.001s
OK
```

Mutation 1: Deleting a line of code

```
def do_graham(p):
    min_index = 0

    for i in range(len(p)):
        if p[i][1] < p[min_index][1]:
            min_index = i

    return min_index
```

```
C:\Users\Admin\Documents\project>python test_convex_hull.py
...F
=====
FAIL: test_y_value_same_x_different (__main__.TestDoGraham.test_y_value_same_x_different)
-----
Traceback (most recent call last):
  File "C:\Users\Admin\Documents\project\test_convex_hull.py", line 16, in test_y_value_same_x_different
    self.assertEqual(do_graham(p), 2)
AssertionError: 0 != 2
-----
Ran 4 tests in 0.001s
FAILED (failures=1)
```

Mutation 2: Inserting a line of code

```
def do_graham(p):
    min_index = 0
    min_index = 1

    for i in range(len(p)):
        if p[i][1] < p[min_index][1]:
            min_index = i
        elif p[i][1] == p[min_index][1] and p[i][0] < p[min_index][0]:
            min_index = i

    return min_index
```

```

C:\Users\Admin\Documents\project>python test_convex_hull.py
...F
=====
FAIL: test_y_value_same_x_different (__main__.TestDoGraham.test_y_value_same_x_different)
-----
Traceback (most recent call last):
  File "C:\Users\Admin\Documents\project\test_convex_hull.py", line 16, in test_y_value_same_x_different
    self.assertEqual(do_graham(p), 2)
AssertionError: 1 != 2
-----

Ran 4 tests in 0.001s

FAILED (failures=1)

```

### Mutation 3: Modifying a line of code

```

def do_graham(p):
    min_index = 0

    for i in range(len(p)):
        if p[i][1] > p[min_index][1]:
            min_index = i
        elif p[i][1] == p[min_index][1] and p[i][0] < p[min_index][0]:
            min_index = i

    return min_index

```

```

C:\Users\Admin\Documents\project>python test_convex_hull.py
FFFF
=====
FAIL: test_basic_case (__main__.TestDoGraham.test_basic_case)
-----
Traceback (most recent call last):
  File "C:\Users\Admin\Documents\project\test_convex_hull.py", line 8, in test_basic_case
    self.assertEqual(do_graham(p), 0)
AssertionError: 1 != 0
=====

FAIL: test_y_value_different (__main__.TestDoGraham.test_y_value_different)
-----
Traceback (most recent call last):
  File "C:\Users\Admin\Documents\project\test_convex_hull.py", line 12, in test_y_value_different
    self.assertEqual(do_graham(p), 1)
AssertionError: 0 != 1
=====

FAIL: test_y_value_edge_case (__main__.TestDoGraham.test_y_value_edge_case)
-----
Traceback (most recent call last):
  File "C:\Users\Admin\Documents\project\test_convex_hull.py", line 20, in test_y_value_edge_case
    self.assertEqual(do_graham(p), 2)
AssertionError: 0 != 2
=====

FAIL: test_y_value_same_x_different (__main__.TestDoGraham.test_y_value_same_x_different)
-----
Traceback (most recent call last):
  File "C:\Users\Admin\Documents\project\test_convex_hull.py", line 16, in test_y_value_same_x_different
    self.assertEqual(do_graham(p), 2)
AssertionError: 0 != 2
-----

Ran 4 tests in 0.002s

FAILED (failures=4)

```

#### iv. Path Coverage

To ensure path coverage where every loop is explored at least zero, one, or two times, we can create the following test cases:

##### Derived Test Cases:

1. **Test Case 1:**  $p = []$  (0 iterations of both loops)
2. **Test Case 2:**  $p = [(1, 1)]$  (0 iterations of the second loop)
3. **Test Case 3:**  $p = [(1, 1), (1, 1), (1, 1)]$  (1 iteration of the first loop, 0 of the second)
4. **Test Case 4:**  $p = [(1, 2), (2, 1), (0, 1)]$  (1 iteration of both loops)
5. **Test Case 5:**  $p = [(1, 2), (0, 1), (0, 1)]$  (2 iterations of the second loop)