

IT314 Software Engineering

Lab 9

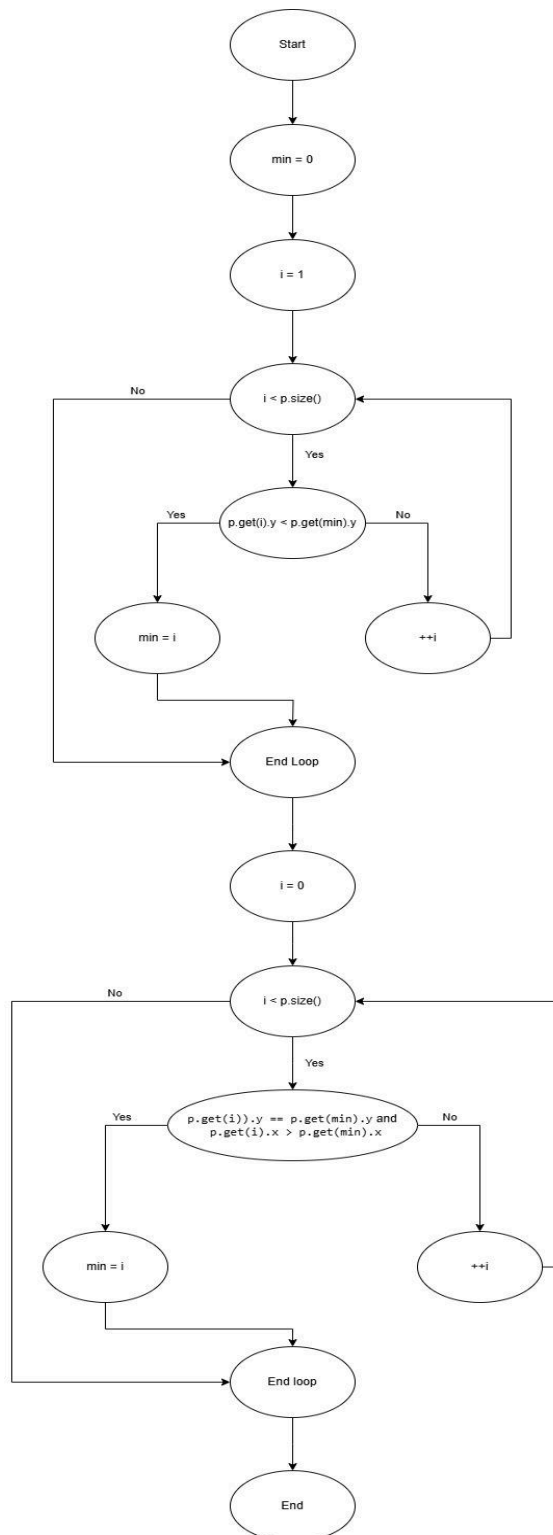
By 202201446

1)

Rewritten code for clarity:

```
Vector doGraham(Vector p) {  
    int i, min;  
    Point t;  
    min = 0;  
  
    for (i = 1; i < p.size(); ++i) {  
        if (((Point) p.get(i)).y < ((Point) p.get(min)).y) {  
            min = i;  
        }  
    }  
  
    for (i = 0; i < p.size(); ++i) {  
        if (((Point) p.get(i)).y == ((Point) p.get(min)).y) &&  
            (((Point) p.get(i)).x > ((Point) p.get(min)).x)) {  
            min = i;  
        }  
    }  
}
```

CFG Graph



2)

Assumptions:

To create meaningful test cases:

- **p** is a **Vector** of **Point** objects, where each **Point** has **x** and **y** components.
- We assume at least two points in **p** to allow comparison in both loops.
- Different **y** and **x** values in **p** enable us to create cases where comparisons yield true and false outcomes.

Test Sets for each Criterion:

a. Statement Coverage

Test: **p** = [Point(x = 1, y = 1), Point(x = 2, y = 0)]

b. Branch Coverage

Test 1: **p** = [Point(x=1, y=1), Point(x=2, y=0)]

Test 2: **p** = [Point(x=0, y=1), Point(x=2, y=1)]

c. Basic Condition Coverage

Test 1: **p** = [Point(x=1, y=1), Point(x=2, y=0)]

Test 2: **p** = [Point(x=1, y=1), Point(x=0, y=1)]

Test 3: **p** = [Point(x=1, y=1), Point(x=2, y=1), Point(x=0, y=0)]

3)

Potential Mutations:

- Mutation 1: Change `((Point) p.get(i)).y < ((Point) p.get(min)).y` to `((Point) p.get(i)).y <= ((Point) p.get(min)).y`.
- Mutation 2: Change `((Point) p.get(i)).y == ((Point) p.get(min)).y` to `((Point) p.get(i)).y != ((Point) p.get(min)).y`.
- Mutation 3: Change `((Point) p.get(i)).x > ((Point) p.get(min)).x` to `((Point) p.get(i)).x >= ((Point) p.get(min)).x`.

4)

| Test Case | Input Points | Loop 1 Iterations | Loop 2 Iterations | Expected min Value |
|-----------|---------------------------------------------------------------------|-------------------|-------------------|--------------------|
| 1 | <code>[(x0, y0)]</code> | 0 | 0 | 0 |
| 2 | <code>[(x0, y0), (x1, y1)], y1 > y0</code> | 1 | 1 | 0 |
| 3 | <code>[(x0, y0), (x1, y1), (x2, y2)], y1 > y0, y2 < y0</code> | 0 | 0 | 0 |
| 4 | <code>[(x0, y0)]</code> | 0 | 1 | 1 |
| 5 | <code>[(x0, y0), (x1, y1)], y0 = y1, x0 < x1</code> | 0 | 1 | 1 |
| 6 | <code>[(x0, y0), (x1, y0), (x2, y0)], x0 < x1 < x2</code> | 0 | 2 | 2 |

Lab Exercise

1)

Control Flow Graph Factory Tool: Yes

Eclipse Flow Graph Generator: Yes

2)

| Test Case | Input p | Expected Result | Coverage Type |
|-----------|----------------|------------------|------------------------------------------------------|
| 1 | [(0,0)] | Minimum is (0,0) | Statement Coverage |
| 2 | [(0,0), (1,0)] | Minimum is (0,0) | Branch Coverage |
| 3 | [(0,1), (0,0)] | Minimum is (0,0) | Basic Condition Coverage |
| 4 | [(1,0), (0,0)] | Minimum is (0,0) | Covers true and false conditions for both conditions |

3)

| Mutation Type | Code Modification | Expected Outcome | Detection by Test Set |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|--------------------------|
| Deletion | Remove $\text{min} = i$; in the loop | Incorrect min point selected | Not detected by TC1, TC2 |
| Insertion | Insert $\text{min} = 0$; at the beginning of the second loop | Resets min point incorrectly | Detected by TC3 |
| Modification | Change $\text{if } (p.\text{get}(i).y < p.\text{get}(\text{min}).y)$ to $\text{if } (p.\text{get}(i).y \leq p.\text{get}(\text{min}).y)$ | Fails to break tie correctly | Not detected by TC2 |

4)

| Test Case | Input p | Expected Result | Explanation |
|-----------|-----------------------|-------------------|---------------------------------|
| 1 | [] | Return empty/null | Loop executes 0 times |
| 2 | [(0,0)] | Minimum is (0,0) | Loop executes 1 times |
| 3 | [(0,1), (1,0)] | Minimum is (1,0) | Loop executes 2 times |
| 4 | [(1,0), (0,0), (2,0)] | Minimum is (0,0) | Covers full loop with ties in y |