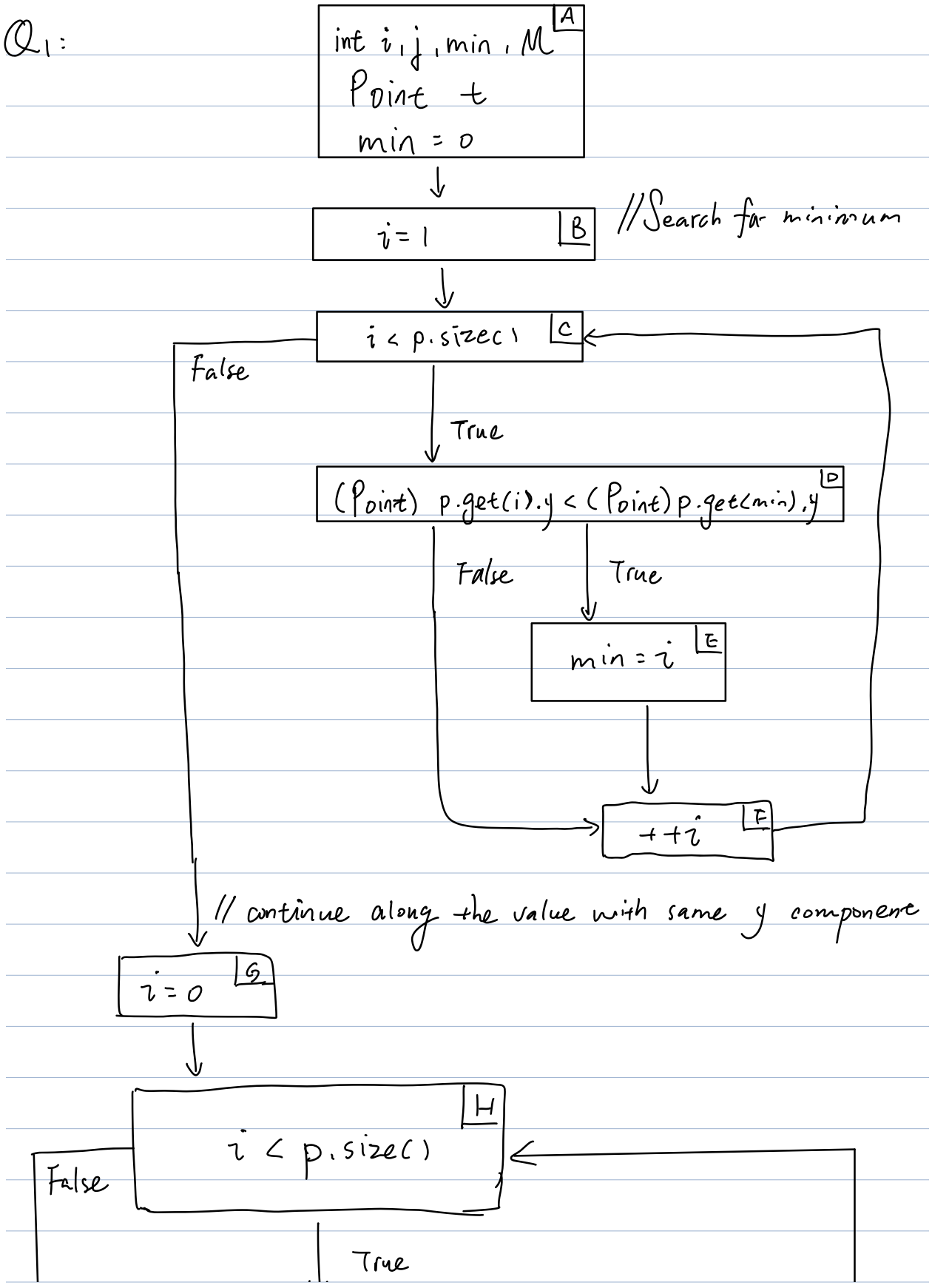
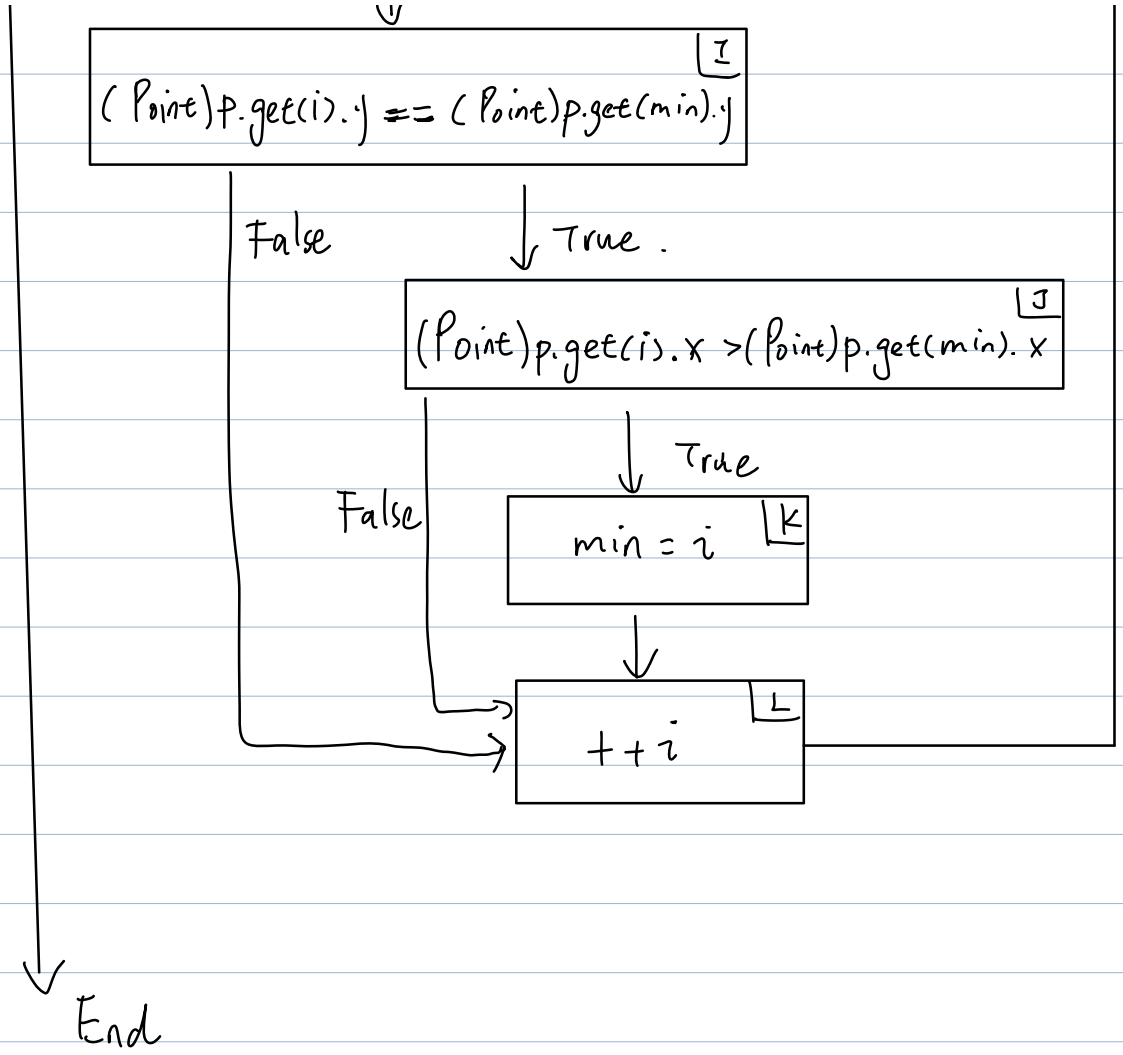
Q1:


A
int i, j, min, M
Point t
min = 0
B
i = 1
//Search for minimum
C
i < p.size() - 1
False
True
D
(Point) p.get(i).y < (Point) p.get(min).y
False
True
E
min = i
F
++i
// continue along the value with same y component
G
i = 0
H
i < p.size()
False
True


I
(Point)p.get(i).y == (Point)p.get(min).y
False
True
J
(Point)p.get(i).x > (Point)p.get(min).x
True
False
K
min = i
L
++i
End

Q3: Statement coverage

Statement coverage 1:

Change int i=1 into 0 in first for loop: (Point) p.get(i).y < (Point) p.get(min).y

Test vector = [1,0], [2,0], [3,0], [0,1], [0,2], [0,3]

Expected result: min = 3

Exact result: min = 3

The mutant statement coverage 1 expected result equal to exact result. So it works

Statement coverage 2:

Change int i=0 into 1 in second for loop

Test vector = [1,0], [2,0], [3,0], [0,1], [0,2], [0,3]

Expected result: min = 3

Exact result: min = 3

It works

However if the test vector change, statement coverage 2. will not work.

Tested vector: [10,0], [2,0], [0,1], [10,5]

then expected result: min = 10

Exact result: min = 2

So it didn't work. it need to be fixed

Branch Coverage 1:

min = 1 instead of min = i

Test Vector	Expected Value	BC1 result
[1,1], [-1,-1]	min = 1	min = 1
[1,1], [-1,-1], [3,-1]	min = 2	min = 1

Coincidentally, it will be correct, otherwise wrong

Branch Coverage 2:

if (Point) p.get(i).y == (Point) p.get(min).y ||
(point) p.get(i).x > Point.p.get(i).x

Test Vector	Expected Value	BC2 result
[1,3], [0,3]	min = 0	min = 1
[1,1], [-1,1]	min = 1	min = 1

It will be wrong in all the situation like [1.3], [0.3]

Question 3:

Test set for "for every loop is explored zero"

Test set : [ ]

Test set for " for every loop is explored once"

Test set : [1, 1], [-1, -1]

Test set for " for every loop is explored two times"

Test set : [1, 1], [-1, -1], [2, 2], [-2, -2], [3, 0], [0, 3]