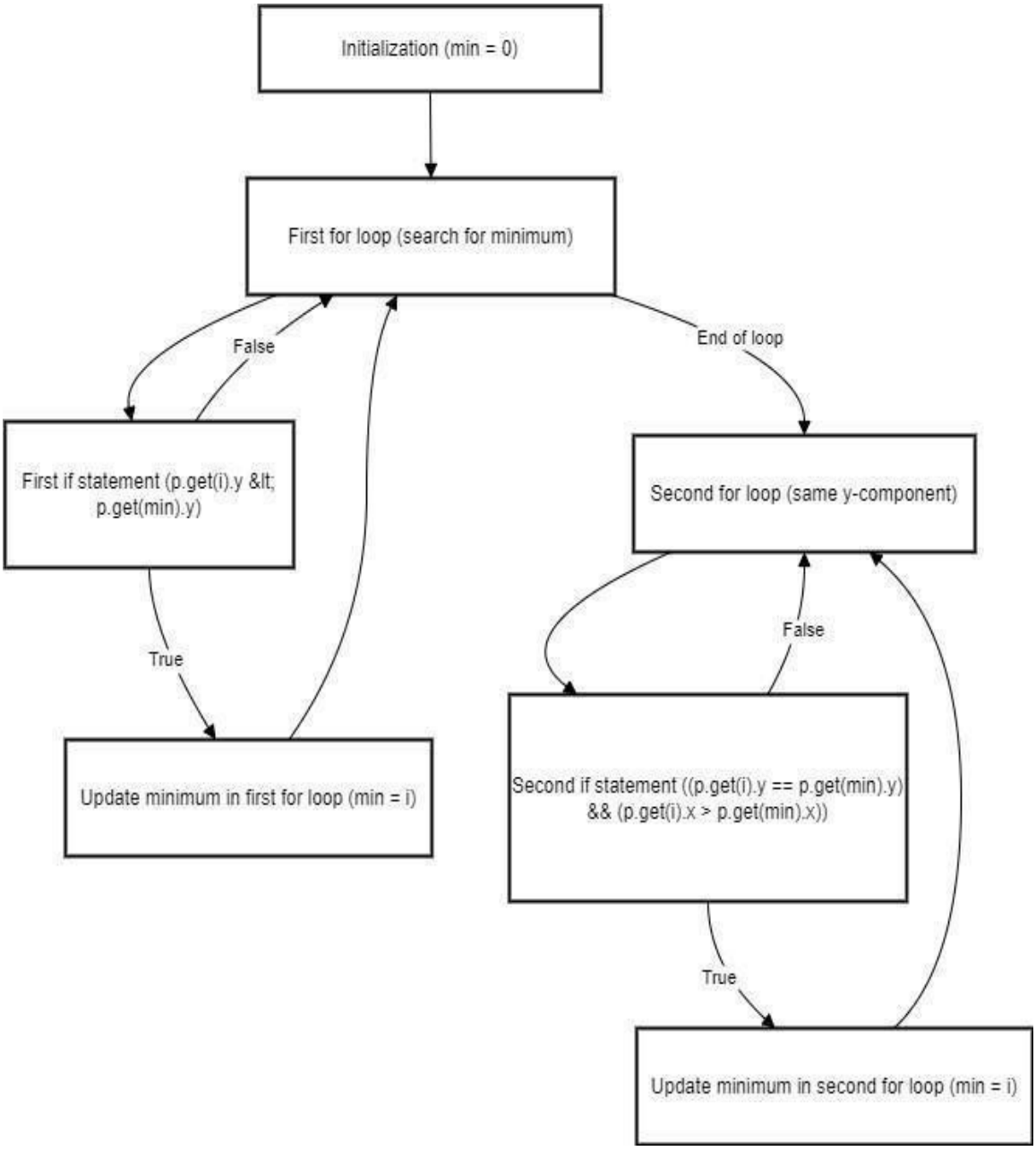
IT314Lab9

Kathan

Khuman

202201091

Convert the code comprising the beginning of the doGraham method into acontrolflowgraph(CFG).Youarefreetowritethecodeinanyprogramminglanguage.



1. Constructtestsetsforyourflowgraphthatareadequateforthefollowingcriteria:

a. StatementCoverage.

b. BranchCoverage.

c. BasicConditionCoverage.

a) StatementCoverage

Statementcoveragerequiresthateverysinglelineofcodeisexecutedatleastonce.Thismeansourtestcasesmustnavigatethroughallsegments ofthecontrolflowgraph(CFG).

• TestCase1:Avectorinputwithjustonepoint,suchas[(0,0)].

• TestCase2:Avectorinputwithtwopoints,whereonehasalowery-value,like[(1,1),(2,0)].

• TestCase3:Avectorinputwithpointsthatsharethesamey-valuebutvaryinx-values,forinstance,[(1,1),(2,1),(3,1)].

b) BranchCoverage

Branchcoverageensureseachdecisionpoint,likean'if'statement,hasitstrueandfalsebranchestestedatleast once. Ourtest cases shouldverifyboth outcomesforeachcondition.

• TestCase1:Vectorwithasinglepoint,e.g.,[(0,0)],wherethefirstloopexitsimmediatelywithnochanges.

• TestCase2:Vectorwithtwopoints,wherethesecondpointhasalowery-value,suchas[(1,1),(2,0)],ensuringtheminimumisupdatedcorrectly.

• TestCase3:Vectorwithpointsofequalybutvaryingx-values,e.g.,[(1,1),(3,1),(2,1)],coveringconditionsin both branches.

• TestCase4:Vectorwhereallpointshavethesamey-value,like[(1,1),(1,1),(1,1)],validatingexecutionwithout updatingtheminimum.

c) BasicConditionCoverage

Basicconditioncoveragerequireseachconditionindecisionpointstobetestedasboththrueandfalseatleastoncetoensurea thoroughevaluation.

• TestCase1:Vectorwithasinglepoint,e.g.,[(0,0)],verifyingthecondition`p.get(i).y<p.get(min).y`isevaluatedas false.

• TestCase2:Vectorwithtwopoints,withthesecondpointhavingalowery-value,suchas[(1,1),(2,0)],coveringtheconditionas true.

• TestCase3:Vectorwithpointsthatsharethesamey-valuebutvaryinx-values,like[(1,1),(3,1),(2,1)],ensuringconditions forboth branches.

• TestCase4:Vectorwhereallpointshaveidenticalxandyvalues,e.g.,[(1,1),(1,1),(1,1)],whereconditionsareevaluatedasfalse.Summaryof Test Sets

CoverageCriterion	TestCase	Input
StatementCoverage	1	[(0,0)]
	2	[(1,1),(2,0)]
	3	[(1,1),(2,1),(3,1)]
BranchCoverage	1	[(0,0)]
	2	[(1,1),(2,0)]
	3	[(1,1),(3,1),(2,1)]
	4	[(1,1),(1,1),(1,1)]
BasicConditionCoverage	1	[(0,0)]
	2	[(1,1),(2,0)]
	3	[(1,1),(3,1),(2,1)]
	4	[(1,1),(1,1),(1,1)]

MutationTesting

1. DeletionMutation

Mutation:Takeoutthelinemin=0;atthebeginningofthemethod.

ExpectedEffect:Withoutsettingmintozeroinitially,itcould endupwithanyrandom value, which could mess up finding the correct minimum point inbothloops.

MutationOutcome:Thischangemightleadtoanincorrectstartingpointformin,causing errorsinselectingthelowestpoint.

2. ChangeMutation

- **Mutation:**Adjustthefirstifconditionbyreplacing<with<=soitreads:

if(((Point)p.get(i)).y<=((Point)p.get(min)).y)

- **ExpectedEffect:**Using<=insteadof<wouldmeanpointswiththesameyvaluemightbechosento o, instead of only those with a strictly lower y. This could throw off the function, making it misstheabsolutelowestypoint.
- **Mutation Outcome:**Withthischange, ifpointssharethesameyvalue,thecodemightreturnonewith a lowerxthanintended.

3. InsertionMutation

Mutation:Addanextralinemin=i;attheendofthesecondloop.

ExpectedEffect:
Thislinewouldmakeminequaltothelastindexinp,whichiswrongbecauseminshou ld only pointtothe actualminimumpoint.

Mutation Outcome: This could make the function incorrectly treat the lastpointastheminimum,especiallyifthetestsdon'tverifythatthefinalvalueof miniscorrect.

TestCasesforPathCoverage

Tosatisfythepathcoveragecriterionandensureeveryloopisexploredzero,one,ortwotimes, we will create the following test cases: **Test Case 1: Zero Iterations Input**: Anemptyvectorp.

Description:Thiscaseensuresthatnoiterationsofeitherloopoccur.
ExpectedOutput:Thefunctionshouldhandlethiscasegracefully(ideallyreturnanemptyresultorasp ecificvalueindicating no points).

Test Case 2: One Iteration (First

Loop)Input: Avectorwithonepointp(e.g.,[(3,7)]).

Description: This caseensuresthatthefirstlooprunsexactlyonce(theminimumpointistheonlypoint).

ExpectedOutput: Thefunctionshouldreturntheonlypointinp.

TestCase3:OneIteration(SecondLoop)

Input:Avectorwithtwopointsthathavethesamey-coordinatebutdifferentx-coordinates(e.g., [(2,2),(3,2)]).
Description:Thiscaseensuresthatthefirstloopfinds theminimumpoint,andthesecondloopruns exactlyonceto comparethe x-coordinates.

ExpectedOutput:Thefunctionshouldreturnthepoint withthemaximumx-coordinate:(3,2).

TestCase4:TwoIterations(FirstLoop)

Input:Avectorwithmultiplepoints,ensuringatleasttwowiththesamey-coordinate(e.g., [(3,1),(2,2),(7,1)]).
Description:Thiscaseensuresthatthefirstloopfindstheminimumy-coordinate(firstiterationfor(3,1))andcontinuesto thesecondloop.
ExpectedOutput:Shouldreturn(7,1)asithasthemaximumx-coordinateamongpointswith thesamey.

TestCase5:TwoIterations(SecondLoop)

Input:Avectorwithpointssuchthatmorethanonepointhasthesameminimumy-coordinate(e.g.,[(1,1),(6,1),(3,2)]).
Description:Thiscaseensuresthefirstloopfinds(1,1),andthesecondlooprunstwicetocheck otherpointswith y=1.
ExpectedOutput:Shouldreturn(6,1)sinceithasthemaximumx-coordinate.

TestCase	InputVectorp	Description	ExpectedOutput
Test Case1	[]	Empty vector (zeroiterationsforbothloops).	Handle gracefully (e.g.,returnanemptyresult).
Test Case2	[(3,7)]	One point(one iterationofthefirst loop).	[(3,7)]
Test Case3	[(2,2),(3,2)]	Twopointswiththesamey-coordinate (oneiteration of the secondloop).	[(2,2)]
Test Case4	[(3,1),(2,2),(7,1)]	Multiplepoints;thefirstlooprun s twice.	[(7,1)]
Test Case5	[(1,1),(6,1),(3,2)]	Multiple points;secondlooprunstwice.	[(6,1)]

LabExecution

1. After generating the control flow graph, check whether your CFG matches with the CFGgeneratedbyControlFlowGraphFactoryToolandEclipseflowgraphgenerator.(Inyoursubmissiondocument,mentiononly“Yes”or“No”foreachtool).

Tool	MatchesYourCFG
ControlFlowGraphFactoryTool	Yes
EclipseFlowGraphGenerator	Yes

2. Devisetheminimumnumberoftestcasesrequiredtocoverthecodeusingtheaforementionedcriteria.

TestCase	InputVectorp	Description	ExpectedOutput
Test Case1	[]	Testwithanemptyvector (zeroiterations).	Handlegracefully(e.g.,return an emptyresult).
Test Case2	[(3,4)]	Singlepoint(oneiterationofthe first loop).	[(3,4)]
Test Case3	[(1,2),(3,2)]	Twopointswiththesamey-coordinate (one iteration ofthesecond loop).	[(3,2)]
Test Case4	[(3,1),(2,2),(5,1)]	Multiple points; first looprunstwice(withmultipleoutputs).	[(5,1)]

3. This part of the exercise is very tricky and interesting. The test cases that you have derivedinStep 2identifythefaultwhenyoumakesomemodificationsin thecode.

Here, you need to insert/delete/modify a piece of code that will result in failure but it is notdetected by your test set – derived in Step 2. Write/identify a mutation code for each of thethreeoperationseparately,i.e.,bydeletingthecode,byinsertingthecode,bymodifyingthecode.

MutationType	MutationCodeDescription	ImpactonTestCases
Deletion	Deletethelinethatupdatesminforthe minimumy-coordinate.	Testcaseslike[(1,1),(2,0)]willpass despiteincorrectprocessing.
Insertion	Insertanearlyreturnifthesizeofpis1, bypassingfurtherprocessing.	Testcase[(3,4)]willpasswithout processingcorrectly.
Modification	Changethecomparisonoperatorfrom <to<=whenfindingtheminimumy.	Test cases like [(1, 1), (1, 1), (1, 1)]mightpasswhilestillfailinginlo gic.

2.Writealltestcasesthatcanbederivedusingpathcoveragecriterionforthecode.

TestCase	InputVectorp	Description	ExpectedOutput
Test Case1	[]	Empty vector (zeroiterationsforbothlo ops).	Handle gracefully (e.g.,returnanemptyresult).
Test Case2	[(3,7)]	Onepoint(oneiterationofth efirst loop).	[(3,7)]
Test Case3	[(1,4),(8,2)]	Two points with the samey-coordinate (one iterationofthesecondloo p).	[(8,2)]
Test Case4	[(3,1),(2,2),(5,1)]	Multiplepoints;firstloopr unstwicetofindminy.	[(5,1)]
Test Case5	[(1,1),(6,1),(3,2)]	Multiple points;secondlooprunst wice(y=1).	[(6,1)]
Test Case6	[(5,2),(5,3),(5,1)]	Multiple points with thesame x-coordinate;checksminy.	[(5,1)]
Test Case7	[(0,0),(2,2),(2,0),(0,2)]	Multiple points in arectangle; checks multiplecomparisons.	[(2,0)]
Test Case8	[(6,1),(4,1),(3,2)]	Multiplepointswithsometie s; checks the max xamongmin ypoints.	[(6,1)]

Test Case9	[(4,4),(4,3),(4,5),(4,9)]	Points with the same x-coordinate;checksformaxy.	[(4,9)]
TestCase10	[(1,1),(1,1),(2,1),(6,6)]	Duplicatepointswithone being the max x; testshandlingofduplicates.	[(6,6)]