Lab 3 Report

黃偉哲
107598019
2019/4/30

1 Test Plan

1.1 Test requirements

The Lab 3 requires to (1) select 6 methods from 6 classes of the SUT (GeoProject), (2) design Unit test cases by using **basis path and graph coverage** technique for the selected methods, (3) develop test scripts to implement the test cases, (4) execute the test scripts on the selected methods, (5) report the test results, and (6) specify your experiences of designing test cases systematically using the ISP technique.

In particular, based on the statement and branch coverage criteria, the **test requirements** for Lab 3 are to design test cases *with* **basis path and/or graph coverage** for each selected method so that "*each statement and branch of the method will be covered by at least one test case* and *the both minimum* ***statement*** *(node) and* ***branch*** *(edge) coverage are 90%, respectively (greater than Lab 2)*".

1.2 Test Strategy

To satisfy the test requirements listed in Section 1, a proposed strategy is to

(1) select **those 3 methods that were chosen in Lab1 or Lab2** and **3 new methods** that are NOT selected previously. The selected methods MUST contain **predicate** and **loop** structures (as many as possible).

(2) set the objective of the minimum statement coverage to be greater than that of Lab 2 and adjust the test objective based on the time available (if necessary).

(3) design the test cases for those selected methods by using the **basis path and graph coverage** testing technique.

1.3 Test activities

To implement the proposed strategy, the following activities are planned to perform.

No.	Activity Name	Plan hours	Schedule Date
1	Study GeoProject	0.5	2019/04/28
2	Learn **basis path and graph coverage**	1	2019/04/28
3	Design test cases for	3	2019/04/30

	the selected methods		
4	Implement test cases	2	2019/04/30
5	Perform tests	1	2019/04/30
6	Complete Lab2 report	2	2019/5/1

1.4 Design Approach

The **basis path and graph coverage** technique will be used to design the test cases. Specifically, the control flow graph (CFG) of each selected method shall be drawn first, and the possible test paths that satisfy the test requirements (i.e., **statement (node) and branch (edge) coverage**) shall be derived from the CFG. The possible **inputs** and **expected outputs** for the derived test paths shall be computed for each selected method. *Add more test cases by considering to satisfy other coverage criteria, such as edge-pair, all-use, or prime-path coverage criteria.*

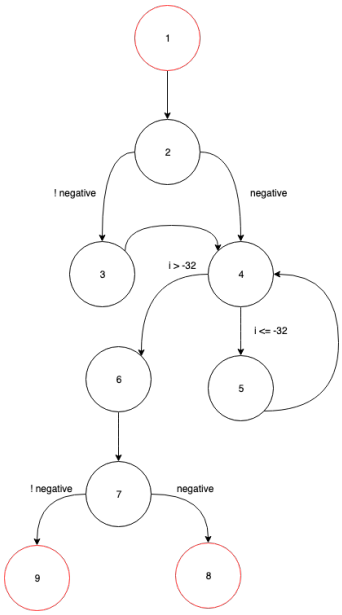
1.5 Success criteria

All test cases designed for the selected methods must pass (or 90% of all test cases must pass) and *both statement and branch coverage should have achieved at least 90%, respectively*.

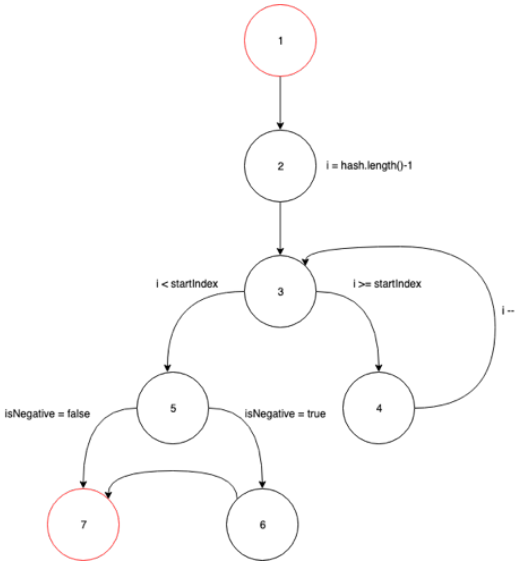
2 Test Design

To fulfill the test requirements listed in section 1.1, the following methods are selected and corresponding test cases are designed.

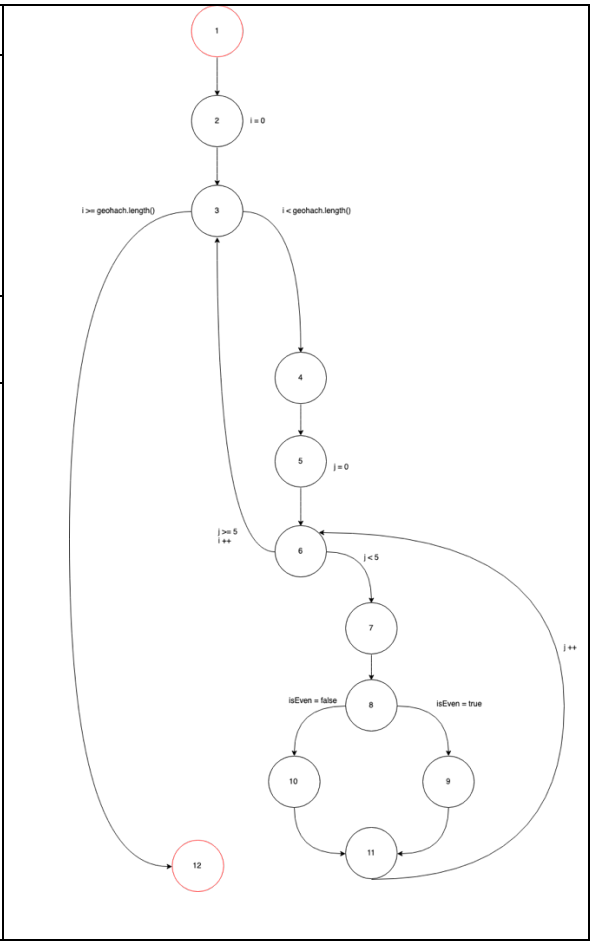
Class : Base32		**Method :** encodeBase32
Basis Path : P1: {n1, n2, n3, n4, n6, n7, n9}, P2: {n1, n2, n4, n5, n6, n7, n8}, P3: {n1, n2, n4, n6, n7, n9}		
No	**Inputs**	**Expected Outputs**
1	i=88888, length=7	"0002qts"
2	i=-89563, length=1	"-2rfv"
3	i=-30, length=7	"-000000y"

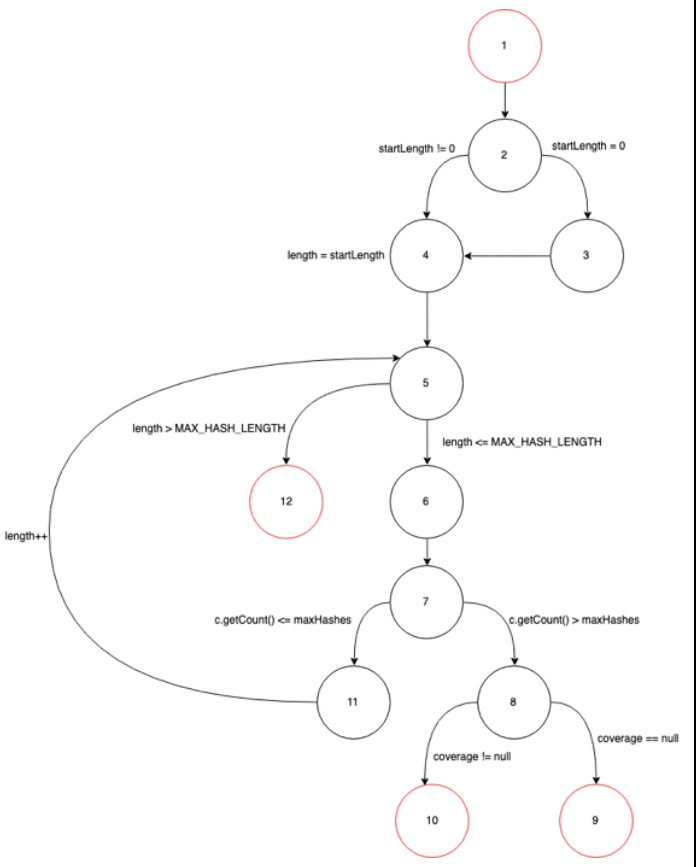


Class : Base32	**Method :** decodeBase32	
Basis Path : P1: {n1, n2, n3, n5, n7}, P2: {n1, n2, n3, n5, n6, n7}, P3: {n1, n2, n3, n4, n3, n5, n6, n7}		
No	**Inputs**	**Expected Outputs**
4	hash="w"	28
5	hash="-j"	-17
6	hash="-29jk"	-75314



Class : GeoHash	**Method :** decodeHash	
Basis Path : P1: {n1, n2, n3, n12}, P2: {n1, n2, n3, n4, n5, n6, n3, n12}, P3: {n1, n2, n3, n4, n5, n6, n7, n8, n10, n11, n6, n3, n12}, P4: {n1, n2, n3, n4, n5, n6, n7, n8, n9, n11, n6, n3, n12}		
No	**Inputs**	**Expected Outputs**
7	geohash="29jw"	LatLong [lat=-38.232421875, lon=-149.58984375]



Class : GeoHash	**Method:** coverBoundingBoxMaxHashes	

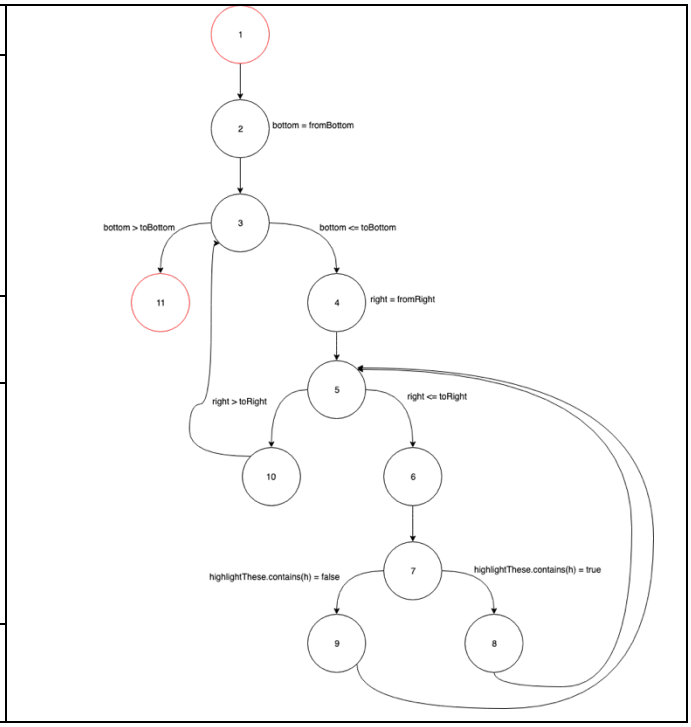
Basis Path :

P1: {n1, n2, n4, n5, n6, n7, n11, n5, n12}

P2: {n1, n2, n3, n4, n5, n6, n7, n8, n9},

P3: {n1, n2, n4, n5, n6, n7, n8, n10},

No	Inputs	Expected Outputs
8	topLeftLat=25.0361156, topLeftLon=121.4639264, bottomRightLat=25.0289061, bottomRightLon=121.4889208, maxHashes=4	"Coverage [hashes=[wsqq7,wsqqk], ratio=21.434198480498]"
9	topLeftLat=1, topLeftLon=-1, bottomRightLat=,-1 bottomRightLon=-1, maxHashes=1	“”
10	topLeftLat=25.0361156, topLeftLon=-121.4639264, bottomRightLat=-25.0361156, bottomRightLon=-121.4639264, maxHashes=2	"Coverage [hashes=[wsqq7vxmfw5y], ratio=Infinity]"

Class : GeoHash	**Method :** gridAsString	

Basis Path :

P1: {n1, n2, n3, n11}

P2: {n1, n2, n3, n4, n5, n10, n3, n11},

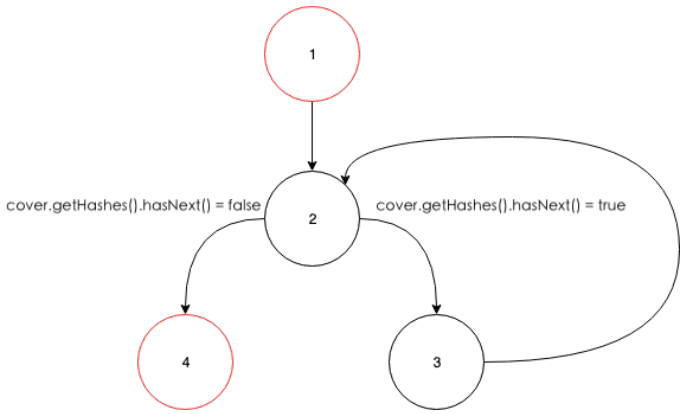
P3: {n1, n2, n3, n4, n5, n6, n7, n8, n5, n10, n3, n11},

P4: {n1, n2, n3, n4, n5, n6, n7, n9, n5, n10, n3, n11}

No	Inputs	Expected Outputs
11	hash="29jw", fromRight=-1, fromBottom=3,toRight=-1, toBottom=1, highlightThese="29jw","29jy", "29jx"	“”
12	hash="29jw", fromRight=1, fromBottom=3, toRight=-1,	“”

	toBottom=1, highlightThese="29jw","29jy", "29jx"		
13	hash="29jw",fromRight=-1, fromBottom=-1,toRight=-1, toBottom=1, highlightThese="29jw","29jy", "29jx"	"29jr \n" + "29jq \n" + "29jm \n"	
14	hash="29jw", fromRight=-1, fromBottom=,-1toRight=-1, toBottom=1, highlightThese="29jq","29jy","29jx", "29jt","29jr","29jm","29jz","29jv"	"29JR \n" + "29JQ \n" + "29JM \n"	

Class : Geomem		**Method :** find
Basis Path : P1: {n1, n2, n4}, P2: {n1, n2, n3, n2, n4}		
No	**Inputs**	**Expected Outputs**
15	topLeftLat=0, topLeftLon=0, bottomRightLat=0, bottomRightLon=0, start=-1000000, finish=100000	"[]"
16	topLeftLat=120, topLeftLon=-120, bottomRightLat=0, bottomRightLon=120, start=-1000000, finish=100000	"[]"



The details of the design are given below:

The Excel file of test cases…

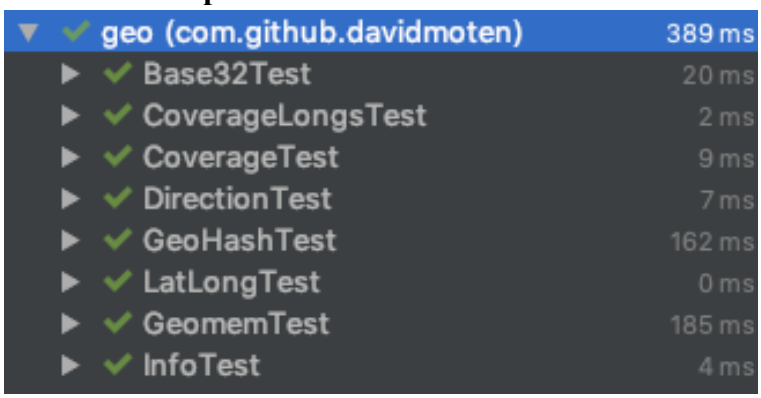
3 Test Implementation

The design of test cases specified in Section 2 was implemented using JUnit 4. The test scripts of 3 selected test cases are given below. The rest of the test script implementations can be found in the link (or JUnit files).

No.	Test method	Source test code
1	public static Coverage coverBoundingBoxMaxHashes(double topLeftLat, double topLeftLon, double bottomRightLat, double bottomRightLon, int maxHashes)	`@Test public void coverBoundingBoxMaxHashes_3(){ Coverage coverage = GeoHash.coverBoundingBoxMaxHashes(topLeftLat: 25.0361156, topLeftLon: -121.4639264, bottomRightLat: -25.0361156, bottomRightLon: -121.4639264, maxHashes: 2); assertEquals(expected: "Coverage [hashes=[3, 9], ratio=Infinity]", coverage.toString()); }`
2	public Iterable<Info<T,R>> find(double topLeftLat, double topLeftLon, double bottomRightLat, double bottomRightLon, long start, long finish)	`@Test public void find_1() { Geomem<String, String> geomem = new Geomem<~>(); geomem.add(lat: 25.0141836, lon: 121.4462453, time: 3, t: "trainStation"); geomem.add(lat: 25.0141836, lon: 121.4462453, time: 14, t: "furniture", id: "ikea"); Iterable<Info<String, String>> it = geomem.find(topLeftLat: 120, topLeftLon: -120, bottomRightLat: 0, bottomRightLon: 120, start: -1000000, finish: 100000); assertEquals(expected: "[]",it.toString()); }`
3	public static LatLong decodeHash(String geohash)	`@Test public void decodeHash() throws Exception{ LatLong result = GeoHash.decodeHash( geohash: "29jw"); LatLong ll = new LatLong( lat: -38.232421875, lon: -149.58984375); assertEquals(ll.getLat(), result.getLat(), delta: 0.001); assertEquals(ll.getLon(), result.getLon(), delta: 0.001); }`

4 Test Results

4.1 JUnit test result snapshot



Test Summary

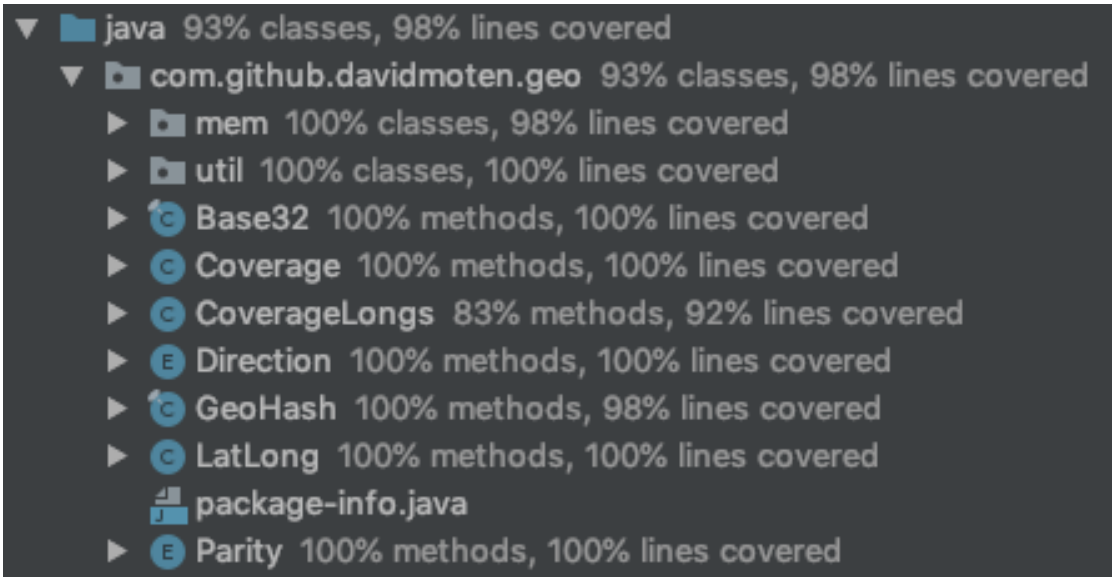
78 tests	0 failures	0 ignored	0.578s duration	100% successful

Packages | Classes

Package	Tests	Failures	Ignored	Duration	Success rate
com.github.davidmoten.geo	68	0	0	0.291s	100%
com.github.davidmoten.geo.mem	10	0	0	0.287s	100%

4.2 Code coverage snapshot

- Coverage of each selected method



- Total coverage

geo

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
com.github.davidmoten.geo		98%		93%	10	149	5	348	0	68	0	10
com.github.davidmoten.geo.mem		97%		70%	6	30	1	61	0	20	0	3
com.github.davidmoten.geo.util		100%		100%	0	4	0	6	0	2	0	1
Total	46 of 2,326	98%	16 of 186	91%	16	183	6	415	0	90	0	14

4.3 CI result snapshot (3 iterations for CI)

- CI#1

README.md

pipeline passed coverage 87%

- CI#2

README.md

pipeline passed coverage 95%

- CI#3

README.md

pipeline passed coverage 97%

- CI Pipeline

Status	Pipeline	Commit	Stages	
passed	#1174 by latest	master 8e8c61e1 Lab3_3		00:01:21 a day ago
passed	#1169 by	master ce850ba4 Lab3_2		00:01:14 2 days ago
passed	#1168 by	master f0da9ddc Lab3_1		00:01:35 2 days ago
passed	#1028 by	master 6e4bcbe7 submit Lab2report and ISP.xlsx		00:01:26 a month ago

5 Summary

從 Lab 3 的練習明顯可以測出更多結果，相較 ISP 似乎更能模擬出實際應用上會發生的問題。