Test Design for Programming Assignment 8 (Separate Tests)
Aman Bhriguvansh

Constructor/BloomFilter(int size):

Goal	Notes	Condition
Code Coverage	All conditions true	Size > 0
Branch Coverage	1 false	Size <= 0
Boundary Coverage	1	Size < 0
Boundary Coverage	1	Size == 0
Boundary Coverage	1	Size > 0
Bad Data	Negative table size or size of 0	Size <= 0
Good Data	Minimum normal configuration	Size == 1

Test Condition	Conditions Satisfied	Assertion
size > 0	CC, BC3	assertNotNull(table[])
size = 0	BC2	assertNull(table[])
size <= 0	B1, BC1, BD	assertNull(table[])
size == 1	GD	assertNotNull(table[])

increment(String address):

Goal	Notes	Condition
Code Coverage	All conditions true	address != null
Bad Data	String is null	address = null
Good Data	Minimum normal configuration	address is empty (“”)

Test Condition	Conditions Satisfied	Assertion
String = “test”	CC	assertTrue(increment(“test”))
String = null	BD1	assertFalse(increment(String)
String = “”	GD1	assertTrue(increment(“”))

reversedString(String address):

Goal	Notes	Condition
Code Coverage	All conditions true	address != null
Bad Data	String is null	address = null
Good Data	Minimum normal configuration	address is empty (“”)

Test Condition	Conditions Satisfied	Assertion
String = “test”	CC	assertEquals(reversedString(“

		test”), “tset”))
String = null	BD1	∅
String = “”	GD1	assertEquals(“”,””)

firstIndex(String address):

Goal	Notes	Condition
Code Coverage	All conditions true	address != null
Bad Data	String is null	address = null
Good Data	Minimum normal configuration	address is empty (“”)

Test Condition	Conditions Satisfied	Assertion
String = “test”	CC	assertEquals(firstIndex(“test”), table[i]) We can calculate i when we set our table size in the method
String = null	BD1	∅
String = “”	GD1	assertEquals(“”,table[i])

secondIndex(String address):

Goal	Notes	Condition

Code Coverage	All conditions true	address != null
Bad Data	String is null	address = null
Good Data	Minimum normal configuration	address is empty (“”)

Test Condition	Conditions Satisfied	Assertion
String = “test”	CC	assertEquals(secondIndex(“test”), table[i]) We can calculate i when we set our table size in the test method
String = null	BD2	∅
String = “”	GD1	assertEquals(“”,table[i])

thirdIndex(String address):

Goal	Notes	Condition
Code Coverage	All conditions true	address has at least 2 letters
Branch Coverage	1 false	normal[] is empty
Branch Coverage	2 false	address is 1 letter
Bad Data	String is null	address = null
Good Data	Minimum normal configuration	address is empty (“id”)

Test Condition	Conditions Satisfied	Assertion
String = “test”	CC	assertEquals(thirdIndex(“test”

		), table[i]) We can calculate i when we set our table size in the test method
String = “”	BC1	assertNull(normal[])
String = “i”	BC2	assertEquals(string.length, string.length%2)
String = null	BD2	∅
String = “id”	GD1	assertEquals(“id”,table[i])

isAnyZero(int first, int second, int third):

Goal	Notes	Condition
Code Coverage	All conditions true	first != 0, second != 0, third != 0
Branch Coverage	1 false	first == 0
Branch Coverage	2 false	second == 0
Branch Coverage	3 false	third == 0
Boundary	1	first < 0
Boundary	1	first == 0
Boundary	1	first > 0
Boundary	2	second < 0
Boundary	2	second == 0
Boundary	2	second > 0
Boundary	3	third < 0
Boundary	3	third == 0
Boundary	3	third > 0

Test Condition	Conditions Satisfied	Assertion
first = 1, second = 1, third =1	CC, B3, B6, B9	assertFalse(first, second, third)
first = 0, second = 0, third =0	BC1, BC2, BC3, B2, B5, B8	assertTrue(first, second, third)
first = -1, second = -1, third = -1	B1, B4, B7	assertFalse(first, second, third)

count(String address):

Goal	Notes	Condition
Code Coverage	All conditions true	address != null
Bad Data	With IPv4 and IPv6, the max length of a string should be 39	Length of address = 45
Bad Data	String is null	address = null
Good Data	Minimum normal configuration	address is empty (“”)
Good Data	Maximum normal configuration	address = 0000:0000:0000:0000:0000:0000:0000:0000

Test Condition	Conditions Satisfied	Assertion
String = “test”	CC	assertEqual(count(“test”),1)
String = null	BD1	∅
String = “”	GD1	assertEqual(conut(“”), 1)

clear():

Goal	Notes	Condition
Code Coverage	All conditions true	table is not an empty set
Branch Coverage	1 false	table is an empty set

Test Condition	Conditions Satisfied	Assertion
Table = [1, 0, 1]	CC	assertEqual(table, {0, 0, 0})
Table = null	BD1	assertNull(table)

Stress Test:

One way we can Stress Test is by generating a list of random IPs. We then add the IPs to a Bloom Filter and check the count. With the way the bloom filter is designed however, returning the count of one insertion might return a value greater than 1 because the hash code % table size is shared with another random IP.