COMS 417 Assignment 1

List the Test cases you have added and for each state:

```
Element † Missed Instructions † Cov. † Missed Branches † Cov. † Missed † Cxty † Missed † Lines † Missed † Missed † Missed † Casses †
                                                            9 13
9 13
                      default =
                                                   45%
                           29 of 65
                                      55% 11 of 20
                                                     45%
                     Total
Coverage Before:
        if((right <=5 && left <= 10 && left >=0) || (right <=10 && left <=5))</pre>
26.
                   if(right <=0)</pre>
27.
28.
                        rslt=-1;
29.
30.
                   else
31.
32.
33.
                   if (right == 0)
34.
35.
                         rslt = 1;
36.
37.
                   else
49.
           public static int inverse (int left, int right)
50.
51.
           //***********************
52.
           // Raises Left to the power of Right
53.
54.
           // precondition : Right >= 0
55.
           // postcondition: Returns Left**Right
56.
57.
            int rslt;
58.
            rslt = right;
59.
             if (right == 0)
60.
                rslt = 1;
61.
62.
63.
            else
64.
65.
                for (int i = 2; i <= left; i++)</pre>
66.
                    rslt = rslt * right;
67.
68.
             return (rslt);
69.
70.
71.
72.
73.
```

These reports tell me that there is not enough testing to go through all of the branches, and test all of the code The inverse method is not tested at all, and there is not enough testing on the power method either

```
@Test public void <u>PowTest3</u>(
 assertEquals(myPow.power(11,0),-1);
```

• This test is testing lines 24 and 26 in Power.java, going through the case that right is ≤ 0 , and that left is > 10

```
@Test public void <u>PowTest4</u>()
  assertEquals(myPow.power(-1,-1),-1);
```

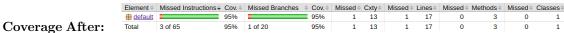
• This test is testing the big if statement on line 24 where both the left and right are negative

```
@Test public void <u>InverseTestl</u>(
  assertEquals(myPow.inverse(1,0),1);
```

• This test is testing the inverse function, going through the branch where right == 0

```
@Test public void <u>InverseTest2()</u>
  assertEquals(myPow.inverse(5,2),32);
```

• This test is making sure that the else statements works on line 63, that the branch returns $right^{left}$



Maximum Coverage

• The maximum coverage is 95%, because there is a line of unreachable code, which is because the if statement on line 26 will always accept when right <= 0, so line 33 will always be covered above on line 26

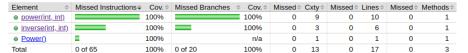
```
26.
                 if(right <=0)</pre>
27.
28.
                    rslt=-1;
29.
30.
                 else
31.
32.
33.
                 if (right == 0)
34.
35.
                      rslt = 1;
36.
37
                 else
```

Two Faults not due to numeric overflow

- 1. the case x^0 will return -1 because line 26 will return -1 if right is zero
- 2. any time where the case x^y where y is negative will fail even though it should return $1/x^y$

Fixing the Faults $\,$ The fault in this program can be fixed by changing the right <= 0 to right < 0

Power



Finding the overflow