CS 4501/6501: Quiz 13

28-Nov-2017

Names:

Instruction: Answer the questions as concisely as you can. Please write neatly; if I can't read it I have to mark it wrong.

There are 2 pages!

Consider the following code:

```
      /**
       * Find last index of element
       *
       * @param numbers array to search
       * @param val value to look for
       * @return last index of val in numbers; -1 if absent
       * @throws NullPointerException if numbers is null
       */
 1.    public static int findVal(int numbers[], int val)
 2.    {
 3.       int findVal = -1;
 4.       for (int i=0; i<numbers.length; i++)
 5.       {
 6.          if (numbers[i] == val)
 7.             findVal = i;
 8.       }
 9.       return findVal;
10.    }
```

Mutation operator: **ROR (Relational Operator Replacement)**

replaces `<, <=, >, >=, ==, !=` with each of the other operators

That is, if the given code has `<` the ROR operator creates 5 ROR mutants by replacing the `<` with the other operators

Apply the ROR operator to line 6 of the given Java method (note: in practice, all lines that are applicable will be mutated. For this quiz, let's focus only on line 6).

You answers must address **all** mutants.

1. (2 pts.) For each mutant, find a test case value that does **not reach** the mutant

Answer:

Test case value: numbers = null
Line 4, numbers.length, throws NPE and thus the mutated line will never be reached
Another possible test case value: numbers = []
Line 4, i<numbers.length, terminates the loop and thus the mutated line will never be reached

2. (8 pts.) For each mutant, find a test case value that **strongly** kills the mutant

Answer:

mutant1: if (numbers[i] < val)
Test case value: [1,2,4,2], 3
Result from running the test
Original: -1, Mutant: 3

mutant2: if (numbers[i] <= val)
Test case value: [1,2,4,2], 3
Result from running the test
Original: -1, Mutant: 3
Another possible test case value: [1,3,4,2], 3
Result from running the test

```
                                        Original: 1, Mutant: 3

mutant3: if (numbers[i] > val)      Test case value: [1,2,4,2], 3
                                    Result from running the test
                                        Original: -1, Mutant: 2

mutant4: if (numbers[i] >= val)     Test case value: [1,2,4,2], 3
                                    Result from running the test
                                        Original: -1, Mutant: 2
                                    Another possible test case value: [1,3,4,2], 3
                                    Result from running the test
                                        Original: 1, Mutant: 2

mutant5: if (numbers[i] != val)     Test case value: [1,2,4,2], 3
                                    Result from running the test
                                        Original: -1, Mutant: 3

Other reasonable test case values that strongly kill mutants are acceptable
```