FIT5171 Applied Session 10
Mutation Testing

Week 10, 2024

> Please do try the questions before coming to the tutorial. Your active participation is the most important!

1. Given a code base with an associated test suite on the code base, if you are to perform mutation analysis on the code base, discuss

    1. What does mutation analysis do?
    2. Why does it do it?
    3. How does it do it?
    4. What is the rationale?

2. The code below shows the `insertSort()` method used in quick sort.

```java
public static void insertSort(int[] A){
  int value = A[1];
  for(int i = 1; i < A.length; i++){
    value = A[i];
    int j = i - 1;
    while(j >= 0 && A[j] > value){
      A[j + 1] = A[j];
      j = j - 1;
    }
    A[j + 1] = value;
  }
}
```

(a) Come up with an *equivalent* mutant by applying a *first-order* mutation. In your answer, identify:

1. The mutation operator applied,
2. The associated statement to be changed, and
3. What the statement is changed to.

(b) Devise a set of three test cases that achieves 100% statement coverage. Come up with three *non-equivalent* first-order mutants of the original program. Determine the kill rate of your test suite on the three mutants.

3. With your assignment group partner(s), find out what mutation operators PIT supports, and what exactly these operators do to the source code.