Course 569
Instructor: Alex Groce
Name: Zehuan Chen

project 2 report
The first tryout of random tester improved algorithm

Introduction

The algorithm I applied is base on the fundamental idea of random test, this algorithm could bring some improvement on current method. In random testing, we consider the system testing as a tree traversal problem, the test case are nodes in this big tree, the algorithms is not randomly pick up a operations or test case in a pool, but with deep traversal first instead. So the the Depth First Search Algorithm that we used to solve searching elements in tree structure is significan useful in this specific situation.

Algorithm

In general thinking, the random tester could keep working till it find the bug, the processing could also be infinity because the bug could be multiple.
The random source is derived from random.Random(), when every time the iteration is done, we shuffle the sequence of sut.state(), the stop condition could be vary, in my program, in order to meet the course requirement, maximun time restrict, depth restrict and width restrict can be applied. When the time you set up is not run out, the testing goes down to the button of the tree using backtrack(), then when the current depth of the traversal bigger than the max depth, the testing iteration is over this round. Using the sut.safely(actions) and test if there are any bug against

Result

`avl0 = avlbug1.AVLTree()`	`# STEP 0`
`val3 = 16`	`# STEP 1`
`val0 = 20`	`# STEP 2`
`avl0.insert(val0)`	`# STEP 3`
`Val0 = 14`	`# STEP 4`
`avl0.insert(val0)`	`# STEP 5`
`avl0.insert(val3)`	`# STEP 6`
`avl0.insert(val2)`	`# STEP 7`

For this project, I using avl tree as my example, the sut.py is generated by tstl, when it runs it show the result like the chart show below.

Conclussion

This is a milstone of my project this term, but I think it still have a lot of problems, the algortihm of feedback-directedd is still hard to implement it all, but the dfs way is the first step to the final, this method shows a good performance that can find bug very quickly, the sample of avlbug.py can be detected.