

# Milestone 1 Reporter

Xiang Li  
932-514-926

## Overview:

From now, I have implement a basic version of a test generation algorithm. My implementation is based on the newCover.py which professor Alex implemented in the class. The newCover.py can be found in:

<https://github.com/agroce/cs569sp16/blob/master/SUTs/newCover.py>

At the beginning of the project 2, I spent a bunch of times to learn TSTL SUT API. This was really difficult to me, because there was no API reference card I can use. So I started with reading the code to understand the functions meanings. During the implementation, I still felt confused about the main algorithm of the paper “Fast Antirandom (FAR) Test Generation”. So I decided to revise the newCover.py first and find a way to understand the TSTL SUT API.

Kai Shi, Yi-Chiao Yang and I are a group. We discuss the algorithm together and help each others to learn the TSTL test generation language.

## Algorithm:

The newCover.py has two separate phases. The phase one uses random test to test the states and their active actions. Meanwhile, the newCover.py can computer the coverage of each branches. So using these accounted data, phase two can randomly test more braches and actions, which can improve coverage rate.

According to the code newCover.py, my algorithm inherits its structure. I also use the two phases structure to build the code. In newCover.py, professor Alex computer the mean number of all actions' cover time. And use this number to find the actions' cover time that below it. In my code, I also divided a test into two phases. Each phase take half of the running time. To compute a benchmark, I was trying to sort the list of action's cover times. And then find the median number of this list. If the actions' whose cover time is below the median number, then grip the state and restore the state in to a new list. In the phase two, randomly run the states that in the new list again to enhance the coverage ratio. Here is the pseudocode of the algorithm:

```
visitedStateList = sort ( vistedStateList )  
coverMedian  = getNumer ( len(visitedStateList) / 2 )
```

```
for s in vistedStateList  
    if vistedStateList < coverMedian:  
        phaseStateList.add(s)  
    else:  
        break
```

## **Future Works:**

For the first version of the code, I have constructed the main structure. This is a rough version, so there are still have some bugs left. In the next step, I will maintain my code, and add more functions into the code.

I still have some confusing about some concepts of the TSTL. For examples, I am still do not understand the different between state and statement clearly. These are the basic concepts. I still need to make sure in rest of class.

In this milestone, my purpose of the project is to adapt to TSTL SUT API. So I imitated the code of newCover.py. Right now, I have learned some frequent used functions and know how to use them. The next step is reading the “Fast Antirandom (FAR) Test Generation” paper carefully and trying to embed this algorithm into to the tester1.py and to improve performance of the random test generation.