Name: Hsi-Min Chou
Student ID: 862254022

CS206 Project Report

Experimental Results

Benchmark	Coverage Criteria	Test Case Prioritization	# of test cases	# of faults exposed
tcas	Statement	Random	4	7
		Total	5	9
		Additional	5	9
	Branch	Random	14	10
		Total	12	10
		Additional	11	10
tootinfo	Statement	Random	7	17
		Total	10	10
		Additional	6	12
	Branch	Random	8	18
		Total	11	12
		Additional	6	12
schedule	Statement	Random	4	2
		Total	6	3
		Additional	3	4
	Branch	Random	10	4
		Total	15	5
		Additional	8	5
schedule2	Statement	Random	5	1
		Total	5	2

		Additional	1	3
	Branch	**Random**	11	3
		Total	13	3
		Additional	6	3
printtokens	**Statement**	**Random**	15	3
		Total	18	3
		Additional	6	4
	Branch	**Random**	17	4
		Total	19	3
		Additional	7	4
printtokens2	**Statement**	**Random**	14	
		Total	17	
		Additional	4	
	Branch	**Random**	20	
		Total	19	
		Additional	9	
replace	**Statement**	**Random**	20	
		Total	17	
		Additional	14	
	Branch	**Random**	25	
		Total	25	
		Additional	23	

Observations

- Additional Coverage can generate the smallest test suites for both statement and branch coverage criteria, which I think might be the best.
- None of the test suites can expose all the faults.
- The generated test suites are small which can make the time become more efficient.
- Random Prioritization is not consistent because when running the program, it changes the order of the test cases every time.
- I'm not able to finish exposing the faults. The reason is that when running the program with the benchmark “replace” and “printtoken2”, it has errors about UnicodeDecodeError.