Homework 1

Implementation of algorithm for Hitting Set using the HGS algorithm

In software testing, test suite reduction is used to reduce the size of a test suite by removing test cases that have redundant coverage. The hitting set problem can be applied to this application.

Given a test suite, TS, a set of requirements, $r_1, r_2, \ldots, r_n$ that must be satisfied to provide testing coverage of a program, and subsets of TS, $T_1, T_2, \ldots, T_n$, with each one associated with each of the requirements such that any one of the test cases t_j belonging to T_i can be used to test r_i, then find a representative set of test cases from TS that satisfies all of the r's.

The literature on the HGS algorithm will help you to solve this problem:

[1] M. J. Harrold, R. Gupta, and M. L. Soffa. A methodology for controlling the size of a test suite. ACM Trans. on Software Engineering and Methodology, 2(3):270–285, Jul. 1993.
[2] Dennis Jeffrey and Neelam Gupta. Improving fault detection capability by selectively retaining test cases during test suite reduction. IEEE Transactions on Software Engineering, 33(2):108–123, 2007.

For this assignment, we provide seven input files in which each test case is followed by a sequence of numbers where each number represents a branch in the code. For instance:

```
0 1 2 3  // Test case 0 covers branches 1, 2, and 3
1 2 1 4  // Test case 1 covers branches 2, 1, and 4
2 2      // Test case 2 covers branch 2
3        // Test case 3 does not cover any branches
```

You implementation should reduce the test suites using the algorithm from [1], specifically using branch coverage.

Submission

Part 1 – Implementation (100%)

1. Code – You should provide your code and if applicable, your makefile. Please zip the files before e-mailing them to our course e-mail address.
2. **Results** –You should provide output files for each input such that each output file contains the reduced test suite. You will also complete the following table with the results from your program:

Input	Original test suite size	Reduced size	Number of unique branches
App 1	300		
App 2	300		
App 3	300		
App 4	250		
App 5	125		
App 6	890		
App 7	169		

Extra credit

1. Implement a GUI for your program that allows users to upload their files, choose the option to reduce the test suite, and displays the reduced test suite and statistics from the table above. (5%)
2. Also implement the algorithm by Jeffrey and Gupta with a slight variation. Use branch coverage as the primary criterion and pairs of branches as the secondary criterion. (25%)

Grading - No credit will be given for code that does not compile. No credit will be given to code that takes longer than 20 minutes to run on any of the above inputs. Your code must be object-oriented and use well-named methods and variables. There will be a 10% penalty if the code is not commented. You can receive 50% credit if you program fully works only for covering arrays, but not mixed-level covering arrays.