CPSC: 480 Software Engineering Exercise 5

11/02/22

Background

Two-sum is a popular interview exercise we explored in lecture 4. The problem is, given an array of integers, and another integer for a target sum, determine if two separate elements in the array add up to the given sum. The problem can easily be solved with a nested loop, but more efficient solutions can be implemented by changing the data structure within the method by sorting or hashing. Some implementations are easier to make mistakes in than others. Unit testing is an effective strategy to prevent and identify simple mistakes like this. In this exercise you will take three implementations of the two-sum method and write unit tests for each. The exercise is built with a custom, lightweight test framework that uses code annotation to track code coverage.

Exercise

1. Clone the GitHub repo at https://github.com/kilgallin/SWEF22-Exercise-6
2. Create a branch named after your UANET Id.
3. Compute the cyclomatic complexity and runtime of each two-sum method and record the result in the provided comment above the function.
4. Write unit tests using the framework provided to verify behavior of the three two-sum methods.
5. Run the tests and check the code coverage and pass rate.
6. For any tests that don't give the correct result on all methods, examine the two-sum method that's failing and identify the cause of the failure.
7. Correct the failure(s) in the method so that all tests pass.
8. Ensure that you have 100% code coverage and 100% pass rate.
9. Commit & push your branch (git push --set-upstream origin <*branch*>)
10. Issue a pull request to main. **Do not commit or merge to main.**

Submission

Submit the PR by **Friday, Nov 4, 11:59 PM**. This will be considered your submission. Grading will be 10% complexity, 20% code coverage * pass rate, 30% bug fixes, and 40% correctness/completeness/appropriateness of unit tests.