

lab9

Brandon Sassano

Tyra Krivonak

Before Examining the Code

- Hash-Table
 - o The key would be the line number of the statement and the value stored inside of the hash would be the code to be run.
- Stack Array
 - o Used for the missing statements, what is apparent is dequeued, so all that is left is a list of the missing statements.
- Queue
 - o Used this to check the missing statements, cover the capacity, which we could use to determine coverage by dividing $\text{length} - \text{num_items} / \text{capacity}$.

Initial Code Examination

When reading the README file, we found it did not provide a useful explanation of the coverage.py file. A large amount of the contents were source codes in other individual files. The source code included testing, however, it seemed as if most were for edge case coverage or formatting. Most of these files had useful comments describing functionality.

When first looking at the source codes it was overwhelming, as there were a very large number of different files. Because a lot of the files and source codes seem connected and co-reliant, finding a starting point was difficult. As we studied the source codes and the rst files, we began to better understand the connections between files in the codebase.

Detailed Code Examination

1. Data.py

- a. Contains the information of executed lines, stores the information and maps co-related information together. The comments in data.py are more useful than the code as every function is extensively described.
- b. Inside of data.py the data type stored is a dict, storing lines, arcs, file_tracers, and runs. These are all stored as keys inside of the JSON data file.
- c. We believe that the code is semi-readable, however there are many built in functions are unfamiliar. Comments are used heavily in the beginning, but are less common afterwards. This code overall seems manageable, simply the length is intimidating.

2. Collector

- a. The collector python file goes through the all of the data that is stored and traces information. The comments in the collector file appeared to be more useful than the code with its descriptions of python attributes and the purpose of functions.
- b. Collector.py used data structures like dictionary and stack. The stack was used for “Collectors” and the program uses a tracer attribute that indicates if the file should be traced. The stack stores “active collectors” which are added when collectors are started and popped when stopped. The data is used to reference specific keys that keep track of mapping file names to dicts with line numbers as keys.
- c. The code is readable but difficult to understand. Comments are written before large functions with detail. We would feel scared about maintaining this code because of the many unfamiliar functions in the program, but, the main structures look like our own.

3. Results

- a. This file computes the code coverage and identifies the missing statements of the tested file. The code appears more useful in this file as comments are minimal.
- b. Results.py uses an array which stores the line numbers of which were covered when testing and which were excluded. Found line numbers are appended to a new list that formats the output of the program which is used later in summary.py for final display.
- c. The quality of the code could be improved, as functions and methods could be explained with more clarity. We would feel uneasy about maintaining this code due to our inexperience with python. However, we still have a good idea about the purpose and structure of each function in results.py as it seems like past coding experiences.

4.Summary

- a. This file provides the summary of the code coverage to readers in a way that shows missed statements. In this file the code is more useful, as there are not many comments provided.
- b. Summary.py uses an array structure to store the information contained within a code, i.e. the lines. The data is appended to a list which will contain some of the basic information contained within the file. The data is used to inform the user of the coverage test results.
- c. We personally could not read this code, as most of it we have not worked with in the past. Comments are used to give a brief summary of the code's functionality. We decided we would not want to maintain this code, as we are not familiar with all the implementations.