CS132: Software Engineering

HW3: Testing

In this homework, we will practice how to generate test cases that cover system execution from different perspectives. There are 5pts in this homework. Please try to answer all the questions in English and submit a .zip file(including your .pdf report and .py code) on Blackboard.

1. Python Unit Test:

Your task is to write unit tests for the **add_book** function provided in **library_system.py**, ensuring that your tests achieve **100% condition coverage**. This means that your tests should focus on the validation of the ISBN input. The add_book function adds a book to a provided dictionary (library), keyed by the book's ISBN, but only if the ISBN is valid.
Here's the function and the exception class you will be writing tests for:

Tips:
0.from library_system import Book, add_book, InvalidISBNException
1.You should initialize library as a dictionary first.
2.Make use of the **assertRaises** method from the **unittest** framework to test scenarios where an InvalidISBNException should be thrown.

library_system.py

```
class InvalidISBNException(Exception):
    """Exception raised for errors in the input ISBN."""

class Book:
    def __init__(self, title, author, isbn):
        self.title = title
        self.author = author
        self.isbn = isbn

def add_book(library, book):
    if not isinstance(book.isbn, str) or len(book.isbn) != 13 or
not book.isbn.isdigit():
        raise InvalidISBNException(f"Invalid ISBN: {book.isbn}")
    library[book.isbn] = book
```

Requirements

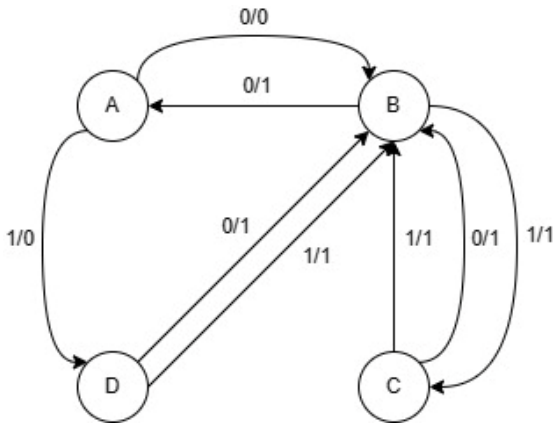
- Write unit tests for the add_book function using Python's unittest framework.(0.5pt)
- Your tests must ensure that **all conditions** within the add_book function are fully covered.(2pts)
- Include comments within your test code to explain which conditions each test case covers.(0.5pt)

2 State Transition Coverage Testing (1.5pts)

A finite state machine with the following state table has a single input X and a single output Y:

Present State	X	Next state	Y
A	**0**	**B**	**0**
A	**1**	**D**	**0**
B	**0**	**A**	**1**
B	**1**	**C**	**1**
C	**0**	**B**	**1**
C	**1**	**B**	**1**
D	**0**	**B**	**1**
D	**1**	**B**	**1**

a. Sketch the state diagram(1pt)



b. List **all** 1-switch test cases that start with state **A** for this state machine (1pts)

Test case	1	2	3	4
Start state	A	A	A	A
Input 1	0	0	1	1
Output 1	0	0	0	0
Next state	B	B	D	D
Input 2	0	1	0	1
Output 2	1	1	1	1
Final state	A	C	B	B
Test coverage item	1	2	3	4

Note: The x/y on each transition refers to input/output of the transition. Also each test case should contain: start state, input(s),next state(s), expected output(s), finish state.