Lab 5: Python Debugging: Debugging a Sub-String Generator

Patrick Di Salvo

November 17, 2019

Due Date: Friday November 22nd 11:59 pm

1 General Overview

The purpose of this lab is to get you to move from testing to debugging. Therefore I have given you a program with several bugs in it, that is not fully implemented. Your job is to implement the program correctly to get the desired result. The program consists of two files, a main file and a file with a class called BinaryString. Your job is to fix two of the three methods in the class BinaryString. The main is already given to you, so do not touch main. You only need to fix the methods in BinaryString called genAllBinStrings and genAllSubStrings. You will note that when you run main before you fix any of the code, the display to the terminal is incorrect. This is because genAllBinStrings and genAllSubStrings are incorrect.

1.1 Overview of the Program

What this program is supposed to use the length of a human readable string, such as "dog" in order to generate all binary strings of length equal to the length of the human readable string. In the case of "dog", that would mean all binary strings of length 3.

The program also takes in a K parameter which is meant to represent "all sub-strings of length K". With the K value you are to print all binary strings of length N with exactly K 1's in them and print all sub-strings of the human readable string of length K.

With the case of "dog", the program must output the binary strings 011, 101 and 110 as well as the sub-strings 'og', 'dg' and 'do'.

2 Method 1: genAllBinStrings

This method is meant to generate all binary strings of length N, save them in an array of binary strings called binaryStrings and print only the binary strings with K 1s in them.

Things NOT to change: The parameters to the method, the return statement in the method, the types of any of the variables in the class or the method

Things to change: The logic inside the method so that the method works correctly.

2.1 Example of a correct result

Given an N value of 3, a K value of 2 and a count value of 0, genAllBinStrings should produce the following:

genAllBinStrings(3, 2, 0): Output:

$$\begin{bmatrix} 0 & 1 & 1 \end{bmatrix}$$
$$\begin{bmatrix} 1 & 0 & 1 \end{bmatrix}$$
$$\begin{bmatrix} 1 & 1 & 0 \end{bmatrix}$$

The following binary strings must be saved in the variable binaryStrings:

$$\begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$$
$$\begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$$
$$\begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$$
$$\begin{bmatrix} 0 & 1 & 1 \end{bmatrix}$$
$$\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$
$$\begin{bmatrix} 1 & 0 & 1 \end{bmatrix}$$
$$\begin{bmatrix} 1 & 1 & 0 \end{bmatrix}$$
$$\begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$$

2.2 Method 2: genAllSubStrings

This method is meant to print all sub-strings of size K from a string of length N. The method calls genAllBinStrings because you are to use the binary strings generated from genAllBinStrings as a way to generate all the sub-strings of size K.

Things Not to Change:The parameters to the method, the return statement in the method, the types of any of the variables in the class or the method, the line of code in the method that looks like this:
self.genAllBinStrings(len(string), k, 0) (line 44 in the file)

Things to change: Any other logic in the method in order to get the method to work correctly.

2.3 Example of a correct result

Given a K value of 2 and a string value of "abc" the method should display to the terminal the following:

genAllSubStrings(2, "abc"): Output:

$$\begin{bmatrix} 0 & 1 & 1 \end{bmatrix}$$
$$\begin{bmatrix} 1 & 0 & 1 \end{bmatrix}$$
$$\begin{bmatrix} 1 & 1 & 0 \end{bmatrix}$$

bc
ac
ab

Note: I am aware the substrings are misaligned with the matrix. Ignore the misalignment, it is not meant to be misaligned.

The reason genAllSubStrings prints the binary strings is because it calls genAllBinStrings and genAllBinStrings prints all binary strings of length N with K 1's in them. This is correct, do not change this.

3 Method: countOnes

This method is meant to help you count the number of '1's' in a binary string. It is correct and does not need to be changed.

4 Grading

The method generateAllBinStrings is worth 1%. It is all or nothing, there are no part marks.
The method generateAllSubStrings is worth 1%. It is all or nothing, there are no part marks.
In your main you will see the following function calls:
generator.genAllSubStrings(2, "abc")
generator.genAllSubStrings(2, "Mary")
generator.genAllSubStrings(3, "Alien")

The first call must produce the result:
['0', '1', '1']
['1', '0', '1']
['1', '1', '0']
bc
ac
ab

The second call must produce the result:
['0', '0', '1', '1']
['0', '1', '0', '1']
['0', '1', '1', '0']
['1', '0', '0', '1']
['1', '0', '1', '0']
['1', '1', '0', '0']
ry
ay
ar
My
Mr
Ma

The third call must produce the result:
['0', '0', '1', '1', '1']
['0', '1', '0', '1', '1']
['0', '1', '1', '0', '1']
['0', '1', '1', '1', '0']
['1', '0', '0', '1', '1']
['1', '0', '1', '0', '1']
['1', '0', '1', '1', '0']
['1', '1', '0', '0', '1']
['1', '1', '0', '1', '0']
['1', '1', '1', '0', '0']
ien
len
lin
lie
Aen
Ain
Aie
Aln
Ale
Ali

If you get this output, you get 100% **UNLESS** you hard code the answer, in which case you will get 0%.

5 Submission

See all other labs for a correct submission.