

### Implementation of Fitness Score function:

Here we have used DDU (Density Diversity Uniqueness) Matrix.

Activity matrix is already given

Density = Sum of all elements of activity matrix /total number of elements of activity matrix

Uniqueness=  $\frac{\text{Total no.of unique components (Unique Columns in Activity Matrix)}}{\text{total number of components}(\text{total number of columns in activity matrix})}$

Diversity:

1. First calculate frequency of each testcase in testsuite as  $n_k$ .
2. Then calculate total number of test case as  $N$
3. Apply following formula to get diversity :

$$\delta = 1 - \frac{\sum_{k=1}^l |n_k|(|n_k|-1)}{N(N-1)}$$

Fitness score =(1-abs(1-2\*density))\*Uniqueness\*Diversity

### Implementation of Suspiciousness function:

Here we calculate Suspiciousness of each component(column of activity matrix)

First Calculate  $C_f$ ,  $C_p$ ,  $N_f$ ,  $N_p$

$C_f$	Number of failing tests that execute C
$C_p$	Number of passing tests that execute C
$N_f$	Number of failing tests that do not execute C
$N_p$	Number of passing tests that do not execute C

Then Apply Ochiai Metric Formula to get the suspiciousness of each component.

$$Ochiai(C) = \frac{C_f}{\sqrt{(C_f + N_f) \cdot (C_f + C_p)}}$$

### Implementation of getRankList function:

Here we first call suspiciousness() function for each component and based on suspiciousness score we assigned them rank and return the list

### Assumption:

1. My component count starts from Zero like if 5 components then it is C0,C1,C2,C3,C4(as mention in the Assumption section of Assignment)
2. The code assumes that the input data, activity\_mat, is a 2D matrix (list of lists) where each row represents a test case or a set of activities, and each column represents a different aspect or component of the test case. It is assumed that the elements are numeric (e.g., integers) specifically activity matrix is a binary matrix.
3. The code assumes that activity\_mat is a matrix with consistent dimensions, meaning that all rows have the same number of columns. This assumption is critical for calculating metrics like Density, Uniqueness, and Diversity.

### Limitation:

1. First Limitation is that it totally depends on optimised activity matrix and output(ranks) may change when we run the same code again because activity matrix changed but the buggy component will get top rank in both the cases.
2. The code assumes that the variables self.errorVec and column\_data contain binary values (0 or 1). It may not work correctly if these variables contain other types of data.

3. The code does not contain error handling or validation for potential issues such as division by zero or unexpected data formats. Example :-We can encounter division by zero error if total elements of activity\_mat is 0 and if number of columns in activity matrix is 0 etc.

## Five Interesting Test Cases:

1. sbfl2.tl and sbfl2\_buggy.tl (with 2 variable x,y)

```
Following List contain components according to their rank(rank 1 is the highly suspicious to be buggy)
[['C5', 1], ['C2', 2], ['C6', 2], ['C0', 3], ['C1', 3], ['C3', 4], ['C4', 4], ['C7', 4]]
DONE..
```

2. s3.tl and s3\_buggy.tl (with 3 variable x,y,z)

```
Following List contain components according to their rank(rank 1 is the highly suspicious to be buggy)
[['C2', 1], ['C3', 1], ['C4', 1], ['C5', 1], ['C6', 1], ['C7', 1], ['C0', 2], ['C1', 2], ['C8', 2], ['C9', 3], ['C10', 3], ['C11', 3], ['C12', 3], ['C13', 3]]
DONE..
```

3. s4.tl and s4\_buggy.tl (with 2 variable x,y)

```
Following List contain components according to their rank(rank 1 is the highly suspicious to be buggy)
[['C6', 1], ['C2', 2], ['C3', 2], ['C7', 2], ['C8', 2], ['C9', 2], ['C11', 2], ['C0', 3], ['C1', 3], ['C4', 4], ['C5', 4], ['C10', 4], ['C12', 4], ['C13', 4], ['C14', 4]]
DONE..
```

4. s5.tl and s5\_buggy.tl (with 2 variable x,y)

```
xclass list
Following List contain components according to their rank(rank 1 is the highly suspicious to be buggy)
[['C4', 1], ['C5', 1], ['C13', 2], ['C14', 2], ['C15', 2], ['C19', 3], ['C20', 3], ['C25', 3], ['C0', 4], ['C1', 4], ['C2', 4], ['C3', 4], ['C6', 4], ['C9', 4], ['C12', 4], ['C16', 4], ['C18', 4], ['C21', 4], ['C24', 4], ['C10', 5], ['C11', 5], ['C7', 6], ['C8', 6], ['C17', 6], ['C22', 6], ['C23', 6]]
DONE..
```

5. s6.tl and s6\_buggy.tl (with 3 variable x,y,z)

```
Following List contain components according to their rank(rank 1 is the highly suspicious to be buggy)
[['C0', 1], ['C1', 1], ['C2', 2], ['C9', 2], ['C3', 3], ['C6', 3], ['C7', 3], ['C8', 3], ['C10', 3], ['C4', 4], ['C5', 4]]
DONE..
```