CS 562
Jingyuan Xu
The library which I want to test named *Examine*, and you can download from this link: https://github.com/Jazzer360/python-examine. This library is the structure parser that allows users to easily see the resulting structure of python object. When we deal with a dataset, we need to make sure the dataset is in the right data type, data structure, and understand the relationship between the metadata and the dataset, in order to get the right result.
Thus, I will use TSTL to test this library. This library which I want to task can help users realize the benefits mentioned above, and it has functions with the usages show as follows:

1 Initialize a structure structure:Takes the type of the value, and recursively creates sub-structures as children if the type is a list, tuple, or dict.
2 Main structure function: An object that reflects the structure of any python data structure.
3 Add structures to get a new structure that is common to both: The returned structure is a newly created structure that reflects the similarities and differences between the two structures added.
4 representation of the underlying structure
5 Copy function: Copies an existing structure and all of it's children
6 generation function: Returns the number of ancestors that are dictionaries
7 string format function: Returns a string representing the type of the structure

A simple useage like this:
we call the analysis function and output the data relation with a dataset. Such like:
input:

```
examine.Structure({'key1': {'subkey1': 'subvalue'}})
```

output:

```
dict
  key1 - dict
    subkey1 - str
```

Then I will use different kind of test cases to test the structure like the example. For example, if the input dataset like this: [{'key1': {'subkey1': 'subvalue'}}], what kind of structure it will output. The correct format output should like this:

```
[dict]
  key1 - dict
    subkey1 - str
```

if we want to test the dataset has the parallel data structure like this: {'key1': 'val1'}, {'key2': 'val2'}
The correct format output should like this:

dict

key1- dict

subkey1 - str

Then I can test different types of the input dataset.

such as if the underlying value for a key is none, checking the kind of data type that this library will output. For clearly readable, this library will not output the none data.

Also, if one dataset contains muitable data type in a list, checking the output from the library. In this suitation, we should get the result “<mixed-type>”.

we may have the data called Tuples. They are typically used with non-homogeneous values, and this library will show a useful information about the contents of tuples. Since a tuple may contain multiple dicts, and each may have it's own structureThe design of this library does not automatically give the structure of dicts that are part of a tuple. If users want to find out those structures, users must do so manually.

For example, we have this dataset

atuple = ({'key': 'val'}, {'anotherkey': 'anotherval'})

and we know the structure of this dataset is (dict, dict)

So the first value of “atuple” is a key, with the string type. The structure like this:

dict

key - str

This is a useful library that can help users understand the dataset which they need to analysis, and help them to check the data type, in order to avoid data errors happen.