

# FE-520 Assignment 2

Dan Wang, Zhiyuan Yao

October 5, 2020

## Submission Requirement:

For all the problems in this assignment you need to design and use Python 3, output and present the results in nicely format.

In your python file, you need contain both function and test part of function. You need to write your code into function with name provided (**wrong name will get 0 points**), and test your function in test block (`if __name__ == "__main__":` . You are strongly encouraged to write comment for your code, because it is a convention to have your code documented all the time.

A template has been provided, you can write your code in this template and use the samples in test block to test your functions. A correct submission should produce the following output in your terminal:

Q1

Correct/Total: 3/3

Q2

The most frequent key is light , its frequency is 3.

The most frequent key is cook , its frequency is 6.

Correct/Total: 2/2

Q3

Correct/Total: 3/3

Q4

Correct/Total: 3/3

Python script must be a '.py' script, Jupyter notebook '.ipynb' is not allowed. You don't need to submit pdf report this time. Do NOT copy and paste from others, all homework will be firstly checked by plagiarism detection tool.

## 1 Loop & Condition Practice(30 pts)

A ball falls freely from a certain height. After each landing, it bounces back to 1/4 of its original height, then it falls again. Define a function(name: Ball\_Drop) with two

arguments, one is the original height, another one is the number of landing (Set this is a default value with 5). Return the meters of Total traveling distance when it bounces after the number of specific landing.

Example:

Input:

OriginalHeight = 100

NumLanding = 2

Total traveling distance =  $100 + 25 + 25 + 6.25 = 156.25$

## 2 String Practice(30pt)

You are required to write a function(name: Get-Token) which can take a string as its argument, and return a dictionary. 1) This function will split this string by blank space into a list of sub-string, 2) then turn each element in this list into lower case without any punctuation (including special characters like '\n', '\t', etc.) and empty string. 3) if a token starts with or ends with a punctuation, remove the punctuation, e.g. "world!" to "world", "'hello'" to "hello". (Hint: you may need to use string.punctuation method by importing string). 4) After this, compute the frequency of each element in this list, and make a dictionary whose key is the element (sub-string) and value is the frequency. 5) Use string formatting to print out a sentence describing the most frequent word and its frequency in this string. 6) Return the dictionary you created in step 4.

Here is an example,

```
>>> myString = '''
    This course is designed for those students have no experience or
    limited experience on Python. This course will cover the basis
    syntax rules, modules, importing packages (Numpy, pandas), data
    visualization, and Intro for machine learning on Python. You will
    need to implement what you learn from this course to do a finance
    related project. This course aims to get you familiar with Python
    language, and can finish a simple project with Python.
    '''
```

```
>>> ret = myFun(myString)
    The most frequent key is 'this', its frequency is 4.
>>> print(ret)
{'this': 4, 'course': 4, 'is': 1, 'designed': 1, 'for': 2,
'those': 1, 'students': 1, 'have': 1, 'no': 1, 'experience': 2,
'or': 1, 'limited': 1, 'on': 2, 'python': 4, 'will': 2,
'cover': 1, 'the': 1, 'basis': 1, 'syntax': 1, 'rules': 1,
'modules': 1, 'importing': 1, 'packages': 1, 'numpy': 1,
'pandas': 1, 'data': 1, 'visualization': 1, 'and': 2,
'intro': 1, 'machine': 1, 'learning': 1, 'you': 3, 'need': 1,
'to': 3, 'implement': 1, 'what': 1, 'learn': 1, 'from': 1,
'do': 1, 'finance': 1, 'related': 1, 'project': 2, 'aims': 1,
```

```
'get': 1, 'familiar': 1, 'with': 2, 'language': 1, 'can': 1,
'finish': 1, 'simple': 1}
```

### 3 Step to Fibonacci sequence((30pt)

Define a function (name: Step\_2.Fibo) that, given an integer  $X$  and a default-value parameter `isIncrement = True`, returns an integer that corresponds to the minimum number of steps required to change  $X$  to a Fibonacci number.

In each step, you can either increase or decrease the current number depending on the parameter `isIncrement`, i.e. you can change  $X$  to  $X+1$  if `isIncrement = True` or  $X-1$  if `isIncrement = False`.

$X$  will be between 0 and 1,000,000 inclusive.

The Fibonacci sequence is defined as follows:

```
F[0]=0
F[1]=1
for each i >= 2: F[i]=F[i-1]+F[i-2]
```

The elements of the Fibonacci sequence are called Fibonacci numbers. The following sequence shows some elements at the head of this sequence:

$$[0, 1, 1, 2, 3, 5, 8, 13, 21, \dots]$$

For example, for  $X=15$  and `isIncrement=False` the function should return 2, because the closest Fibonacci number on the left is 13, thus  $15-13 = 2$ ; If `isIncrement=True`, then the result should be 6 because  $21 - 15 = 6$ .

For  $X=1$  or  $X=13$  the function should return 0.

### 4 Identity Substring(30pt)

Write a function (name: `identi_Substring`) that, given a string  $S$ , returns an integer that represents the numbers of ways in which we can select a non-empty substring of  $S$  where all of the characters of the substring are identical. Two substrings with the same letters but different in locations are still considered different.

For example, the string "zzzyz" contains 8 such substrings. Four instances of "z", two of "zz", one of "zzz" and one instance of "y". String "k" contains only one such substring: "k".

The length of  $S$  will be between 1 and 100, inclusive.

Each character in  $S$  will be a lowercase letter (a-z).