Mastering Data Structures and Algorithms
In Class Exercise 5

Yiying Peng (Christine)

March 12, 2019

Problem

Create a program for simulating the fortune unix command. The file that contains fortunes uses % in a single line to separate fortune statements. The fortune command mist parse the file and represent in memory and then select at random one statement and print that every time invoked.We assume that the contents of the file fit in memory properly and is not too big.

If empty statement in the fortune file is empty then if that statement is selected at random, an empty string must be printed.

Statements may be multiline and they should be printed verbatim if that is the case.

As an example:

Your fortune is bad today!

%

No Fortune Today! Sorry.

%

Today is a good day...

To play the lottery

%

%

Sorry not much fortune today!

%

There are 5 statements above so your program must parse the file and find that out and store the string in some data structure then select a number between 0 and 4 if 2 is selected the following must be printed:

Today is a good day...

To play the lottery

Sol. My Python algorithm implementation is as follows.

```python
import random


class Solution:
    def parse_file(self, filename):
        with open(filename, 'r') as f:
            lines = f.readlines()
        lines = [line.strip() for line in lines]

        lst = []
        block = []
        for i, line in enumerate(lines):
            if line != '%':
                block.append(line)
            else:
                lst.append(block)
                block = []
        if block:
            lst.append(block)
        return lst

    def fortune_unix(self, filename):
        lst = self.parse_file(filename)
        num = random.randint(0, len(lst) - 1)
        return '\n'.join(lst[num])


def main():
    sol = Solution()
    print(sol.fortune_unix(filename='fortunes.txt'))


if __name__ == '__main__':
    main()
```

The time complexity is $O(n)$, and the space complexity is also $O(n)$. □