Assignment-8
ELP-718 Telecom Software Laboratory

Putlur Hithesh Reddy
2018JTM2244
2018-2020
A report presented for the assignment on Python and github

Bharti School of
Telecommunication Technology and Management
IIT DELHI
India
27-Sep-2018

Contents

1 Objective

The assignment is designed to know about Python and the ease of programming in python. This also introduces to the basic usage of github and its relevance in software development life cycle

2 Problem Statement-1

2.1 Problem Statement

Finding Valid Cross

- IIT Delhi, has just got the strongest computer
- The professors in charge wants to check the computational capacity of the computer
- So, they decided to create the problem which is to be given as an assignment to students
- You have to help the professor to check the computation capability of the computer
- A valid cross is defined here as the two regions (horizontal and vertical) of equal lengths crossing over each other
- These lengths must be odd, and the middle cell of its horizontal region must cross the middle cell of its vertical region
- Find the two largest valid crosses that can be drawn on smart cells in the grid, and return two integers denoting the dimension of the each of the two largest valid crosses
- In the above diagrams, our largest crosses have dimension of 1, 5 and 9 respectively
- The two crosses cannot overlap, and the dimensions of each of the valid crosses should be maximal

2.2 Algorithm and implementation

- n and m are taken from the user and checked for the given constraint
- [1] Strings are taken form the user into a list
- Each string is taken and started from each element and searched for S
- Valid crosses are defined in functions
- Each element of the matrix is sent to the functiona nd checked for valid cross

2.3 constraints

- n lies between 2 and 105
- m lies between 2 and 105

2.4 Input and Output Format

- *Input Format*

  The first line contains two space-separated integers, n and m

  Each of the next lines n contains a string of m characters where each character is either S (Smart) or D (Dull)

  These strings represent the rows of the grid

  If the jth character in the ith line is S, then (i,j) is a cell smart

  Otherwise it's a dull cell

- *Output Format*

  [2]

  Find two valid crosses that can be drawn on smart cell of the grid, and return the dimension of both the crosses in the reverse sorted order(i.e. First Dimension should be the larger one and other should be smaller one)

2.5 Test cases or Sample Inputs and Respective results

- *Sample Input 0*
- 5 6
- SSSSSS
- SDDDSD
- SSSSSS
- SSDDSD
- SSSSSS
- *Sample output 0*
- 5 1
- *Sample Input 1*
- 6 6
- DSDDSD
- SSSSSS
- DSDDSD

- SSSSSS
- DSDDSD
- DSDDSD
- *Sample output 0*
- 5 5 there can be more than one valid crosses of the same dimension, you can consider any
- *Sample Input 2*
- 5 9
- SSSSDSDDD
- DDSDDDDDD
- SSSSSDDDD
- DDSDDSDDD
- DSSSDDDDD
- *Sample output 0*
- 9 1

2.6 Difficulties faced

- Deciding the functions to be used
- Comparing cell blocks with starting element or middle element

3 Problem Statement-2

3.1 Problem Statement

Finding encrypted messades

- After, getting mix results of valid crosses, professors decided to test the computation abilities on one more problem
- This time professors wanted to test the decryption capabilities of the computer
- Encryption of a message requires three keys, k1, k2, and k3
- The 26 letters of English and underscore are divided in three groups, [a-i] form one group, [j-r] a second group, and everything else ([s-z] and underscore) the third group
- Within each group the letters are rotated left by ki positions in the message
- Each group is rotated independently of the other two. Decrypting the message means doing a right rotation by ki positions within each group

3.2 Algorithm and implementation

- All the required inputs are taken from the user
- The inputs are checked for constraints
- Groups given are considered as lists
- Empty lists are defined and element sare stored into them after grouping
- [3]The obtained list are rotated and the respective decrypted message is printed onto the console

3.3 constraints

- Length of the string lies between 1 and 150
- ki lies between 1 and 150 (i=1,2,3)

3.4 Input and Output Format

- *Input Format*

  All input strings comprises of only lowercase English alphabets and underscores

- *Output Format*

  For each encrypted message, the output is a single line containing the decrypted string

3.5 Test cases or Sample Inputs and Respective results

- *Sample Input 1*

  2 3 4

  dikhtkor_ey_tec_ocsusrsw_ehas_

- *Sample Output 1*

  hardwork_is_the_key_to_success

3.6 Difficulties faced

- String to list conversion
- Rotating the list

4 C Programming Codes

```
###### this is the first .py file ###########

####### write your code here ##########

def main():
        n,m=int(input("Enter the integer n: ")), int(input("Enter the integer m: "))
        if not (n>1 and n<=105):
                return
        if not (m>1 and m<=105):
                return


        l= []
        listofmin=[]
        for x in range(n):
                s= list(input())
                l=l+[s]
        print(l)

        i=0 ; j=0
#       a='S'|'D'
        countr,countl,countd,countu=0,0,0,0
        for a in l[i][j]:
                for a in  l[i][j]:
                        while l[i][j]=='S':
                                countr+=1
                                if j==m-1:
                                        break
                                j+=1
                        j=j-countr
                        while l[i][j]=='S':
                                countl+=1
                                if j==-1:
                                        break
                                j-=1
                        j=j+countl
                        while l[i][j]=='S':
                                countu+=1
                                if i==-1:
                                        break
                                i-=1
                        i=i+countu
                        while l[i][j]=='S':
                                countd+=1
                                if i==n-1:
                                        break
```

```
                        i=i-countd
                        j+=1
                i+=1
                listofmin=listofmin+[min(countr,countl,countd,countu)]
                print(listofmin)
                if i==n-1:
                        break


        max_value=max(listofmin)
        if max_value>1:
                print(max_value+" "+max_value)
        else:
                temp = max_value
                listofmin.remove(max_value)
                max_value=max(listofmin)
                print(temp+" "+max_value)




#               for S in l[i][j]
#                       i-=1
#                       while l[i]==S
#                               countl+=1
#               i=i+countl
#               while l[i][j-1]
#                       while l[i-=1][j]
#
#       import re
#
#       for S in l
#
#       maxpattern = 0
#       if (m>)
#
# i=0
# count=0
# while i<n
#       for S in list[i][:1]
#               if count==1
#                       break
#               count+=1
#       for S in list[i][1:2]
#               for S in list[i+1][:j]


if __name__ == '__main__':
```

```
        main()

###### this is the second .py file ############

####### write your code here ##########
#Main function
def main():
        #key inputs
        k1,k2,k3=int(input("Enter the k1 key: ")), int(input("Enter the k2 key: ")), i
        if not (k1>0 and k1<=150):
                return
        if not (k2>0 and k2<=150):
                return
        if not (k3>0 and k3<=150):
                return

        #String input
        S= input("Enter the string required to be decrypted: ")
        print(S)
        length=len(S)

        if len(S)>150:
                exit()

        #groups given in the question
        group1=['a','b','c','d','e','f','g','h','i']
        group2=['j','k','l','m','n','o','p','q','r']
        group3=['s','t','u','v','w','x','y','z','_']

        list1=[]
        list2=[]
        list3=[]
        reqlist=[]

        i=0
        j,k,m=0,0,0

        #loop for separing the string into lsits according to groups defined
        while i<len(S):
                if S[i] in group1:
                        list1=list1+[S[i]]
                        j+=1

                if S[i] in group2:
                        list2=list2+[S[i]]
                        k+=1

                if S[i] in group3:
```

```
                        list3=list3+[S[i]]
                        m+=1

                i+=1

        #function definition for rotation of elements in the grouped lists
        def rightRotate(l, n):
            return l[-n:] + l[:-n]


        list1 = rightRotate(list1,k1)
        list2 = rightRotate(list2,k2)
        list3 = rightRotate(list3,k3)

        i,j,k,m=i-i,j-j,k-k,m-m

        #loop for converting enrypted message into decrypted one
        while i<len(S):
                if S[i] in group1:
                        reqlist=reqlist+[list1[j]]
                        j+=1

                if S[i] in group2:
                        reqlist=reqlist+[list2[k]]
                        k+=1

                if S[i] in group3:
                        reqlist=reqlist+[list3[m]]
                        m+=1

                i+=1

        s= ''.join(reqlist)
        print(s)

if __name__ == '__main__':
        main()
#import numpy
#numpy.roll(list1,k1)
#numpy.roll(list2,k2)
#numpy.roll(list3,k3)

#import collections
#list1.rotate(k1)
#list2.rotate(k2)
#list3.rotate(k3)

#function for rotating the list elements
```

```
#def rotate(list,x):
#       x=x%len(list)
#       return list[x:] + list[:x]

# Returns the rotated list
# def rightRotate(lists,num):
#     output_list = []
#     # Will add values from n to the new list
#     for item in range(len(lists) - num, len(lists)):
#         output_list.append(lists[item])
#     # Will add the values before
#     # n to the end of new list
#     for item in range(0, len(lists) - num):
#         output_list.append(lists[item])
#     return output_list

#function for rotating the list elements
```

5 Screenshots related to coding

References

[1] https://www.tutorialspoint.com/python/python_strings.htm.

[2] https://www.tutorialspoint.com/git/git_basic_concepts.htm.

[3] https://www.atlassian.com/git/tutorials.