ELL 780

Software Lab

INDIAN INSTITUTE OF TECHNOLOGY
DELHI

Assignment 8

September 27, 2018

Name:
Mansi Garg

Entry Number:
2017EET2302

Contents

1 Problem Statement 1

Statement: 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. Can you 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.

1.1 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**

  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).

- **Sample Input** 5 6
  S S S S S S
  S D D D S D
  S S S S S S
  S S D D S D
  S S S S S S

- **Sample Output** 5 1

1.2 Assumptions

- 2 <= n,m <= 105
- Value of n and m should be provided
- Size of matrix input should in accordance of the dimentions
- No ovelapping of cross should be there
- Space is there between each element of a row

1.3 Program Structure

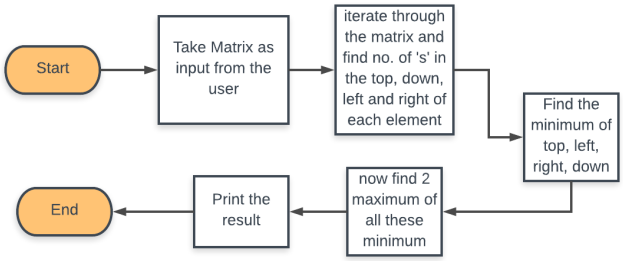

Figure 1: Flow Chart of Problem 1

1.4 Algorithm and Implementations

- The program is written in PYTHON
- Take matrix as input
- Parse through the matrix and count no. of top, down, left and right for each element
- Find min of top, down, left and right
- Find 2 max of these mins

1.5 Screenshots

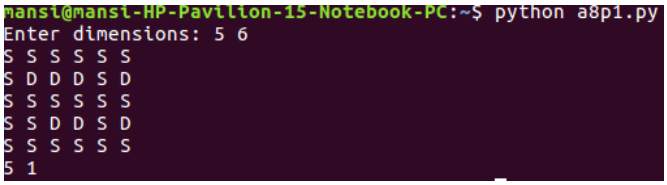

Figure 2: Screenshot 1 of Problem 1

2 Problem Statement 2

Statement: 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

2.1 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.
- **Sample Input** 2 3 4
  dikhtkor_ey_tec_ocsusrsw_ehas_
- **Sample Output** hardwork_is_the_key_to_success

2.2 Assumptions

- 1 <= Length of the string <=150
- 1<= ki <=150 (i=1,2,3)

2.3 Program Structure

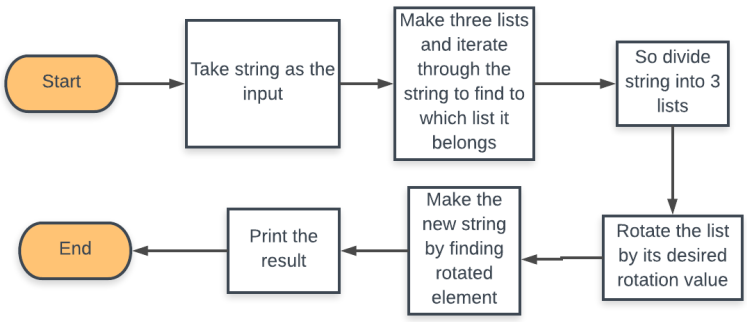

Figure 3: Flow Chart of Problem 2

2.4 Algorithm and Implementations

- The program is written in PYTHON
- Divide the string into 3 lists
- Rotate each list by its desired rotation

- Make a string out of the rotated characters
- Print the string

2.5 Screenshots

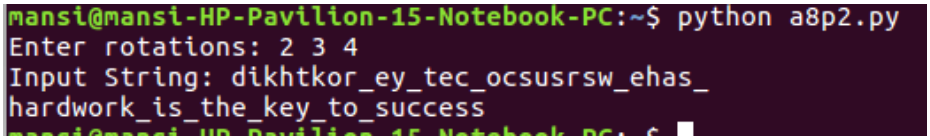

Figure 4: Screenshot of Problem 2

3 Appendix

3.1 Appendix-A: code for ps1

Problem Statement 1

```
#Node that stores 5 things, value, top, down, left and right
class Node:
  def __init__(self, data):
    self.data = data
    self.top = 0
    self.down = 0
    self.left = 0
    self.right = 0
    self.plus = 0

#Take input
x = input("Enter dimensions: ")
n = int(x.split(" ")[0])
m = int(x.split(" ")[1])

#Initialize mtrix
mat = [[0 for x in range(m)] for y in range(n)]
newmat = [[0 for x in range(m)] for y in range(n)]

#Take input in matrix
for i in range(n):
  mat[i]=input().split(" ")

for i in range(n):
  for j in range(m):
    newmat[i][j]= Node(mat[i][j])
maxplus=[]

#Parse through the matrix to find no. of 's' in top, left, down, right
for i in range(n):
  for j in range(m):
    if(newmat[i][j].data == 'S'):
      top = i-1
      down = i+1
      left = j-1
      right = j+1

      while(top>=0 and newmat[top][j].data != 'D'):
        newmat[i][j].top=newmat[i][j].top+1
        top = top-1

      while(down<n and newmat[down][j].data != 'D'):
        newmat[i][j].down=newmat[i][j].down+1
        down = down+1

      while(left>=0 and newmat[i][left].data != 'D'):
        newmat[i][j].left=newmat[i][j].left+1
        left = left -1

      while(right<m and newmat[i][right].data != 'D'):
        newmat[i][j].right=newmat[i][j].right+1
        right = right+1

      #Find min of top, down, left and right
      #this will give length of plus on that element
```

```
56        newmat[i][j].plus = min(newmat[i][j].top,newmat[i][j].down,newmat[i][j].left
      ,newmat[i][j].right)
57
58        maxplus.append(newmat[i][j].plus)
59
60 #Function to get 2 maximum elements
61 def Large(list1):
62     largest = list1[0]
63     largest2 = list1[0]
64     for item in list1:
65         if item > largest:
66             largest = item
67         elif largest2!=largest and largest2 < item:
68                 largest2 = item
69     print((largest*4)+1 , (largest2*4)+1)
70
71 Large(maxplus)
```

3.2 Appendix-B: code for ps2

Problem Statement 2

```
#Initialize th three lists
list1 = ['a','b','c','d','e','f','g','h','i']
list2 = ['j','k','l','m','n','o','p','q','r']
list3 = ['s','t','u','v','w','x','y','z','_']

#Input the rotation value
x = input("Enter rotations: ")
k1 = int(x.split(" ")[0])
k2 = int(x.split(" ")[1])
k3 = int(x.split(" ")[2])

#Input the string
strr = input("Input String: ")

list4=[]
list5=[]
list6=[]

#Divide the string into 3 parts
for ch in strr:
  if ch in list1:
    list4.append(ch)
  elif ch in list2:
    list5.append(ch)
  elif ch in list3:
    list6.append(ch)

#Function to rotate listt by n
def rotate(listt, n):
  rlistt=[]
  for i in range(len(listt)):
    rlistt.append(0)
  for i in range(len(listt)):
    rlistt[i] = listt[((i-n)%len(listt))]
  return rlistt

#Call rotate for all the three lists
rlist4=rotate(list4,k1)
rlist5=rotate(list5,k2)
rlist6=rotate(list6,k3)

i=0
j=0
k=0

#Create new String and print
output=[]
for ch in strr:
  if ch in list1:
    output.append(rlist4[i])
    i=i+1
  elif ch in list2:
    output.append(rlist5[j])
    j=j+1
  elif ch in list3:
    output.append(rlist6[k])
    k=k+1

```

```
59 for ch in output:
60   print(ch,end="")
61 print()
```

References

[1] Python
https://docs.python.org/3/

[2] Github
https://www.atlassian.com/git/tutorials

[3] Github
https://git-scm.com/download/linux