Assignment 8

ELP 780 Software Lab

Varun Sood

2017EET2839

2017-19

A report for the assignment on

Python Programs

Bharti School of
Telecommunication Technology and Management
IIT DELHI
India

September 27, 2018

Contents

1 Problem Statement 1

1.1 Problem statement

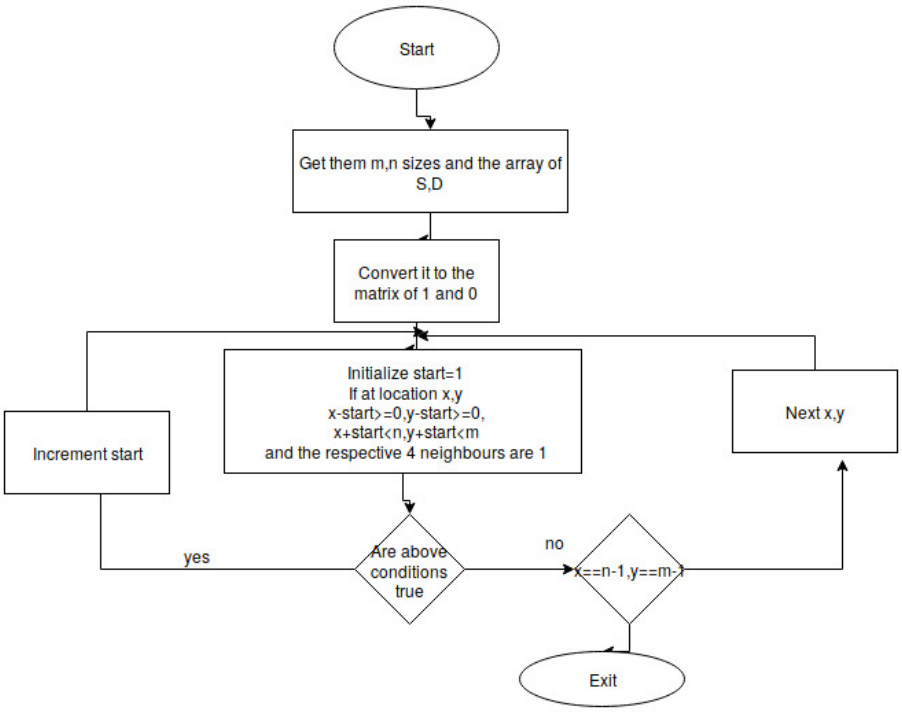
Find the two largest valid crosses that can be drawn on smart cells in the grid, and return two integers denoting the dimension of the each of the two largest valid crosses. In the above diagrams, our largest crosses have dimension of 1, 5 and 9 respectively .

Note: The two crosses cannot overlap, and the dimensions of each of the valid crosses should be maximal.

1.2 Assumptions

- The input file name is taken from the user.
- The file format is given by the user.
- Program is assumed to be run only once and files deleted which were created before running again.

1.3 Program Structure



1.4 Algorithm and Implementation

- First get the elements and 2 d array.
- Generate a matrix of 1 and 0, 1 for the S and 0 for the D.
- Loop over the matrix trying to increase the dimension of the cross in each dimension.
- If true for 1 value of start increase start test again in all 4 directions, up, down , left and right.

1.5 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**
  Print 2 maximums bigger first.

1.6 Test Cases

Input
5 6
SSSSSS
SDDDSD
SSSSSS
SSDDSD
SSSSSS
Output
5 1

1.7 Difficulty/Issues faced

- Python donot define data type at initialization so error might crop up.
- latex is difficult to use
- Python is different then the previous languages..

1.8 Screenshots

```
varun@varun-HP-Pavilion-15-Notebook-PC:~/assignment-8$ python3 ps1.py
Enter two values:  5 6
Enter the listSSSSSS
Enter the listSDDDSD
Enter the listSSSSSS
Enter the listSSDDSD
Enter the listSSSSSS
5 1
```

Final output

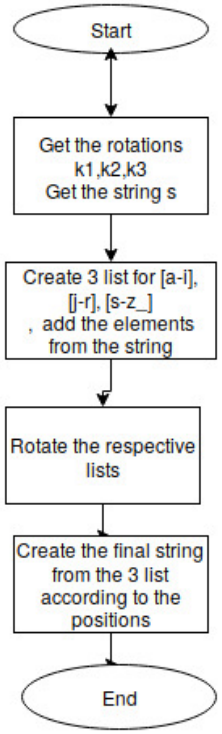
2 Problem Statement 2

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 Assumptions

- The input file name is taken from the user.
- The file format is given by the user.
- Program is assumed to be run only once and files deleted which were created before running again.

2.2 Program Structure



2.3 Algorithm and Implementation

- Input values of rotation k1,k2,k3 are taken from the user.
- Input string is taken from the user.
- Get the different strings out in 3 different lists.
- Rotate them and atlast add them in the string.
- Print the string.

2.4 Input and Output Format

- **Input format**
  k1 k2 k3
  string_sample
- **Output format**
  rotated_string

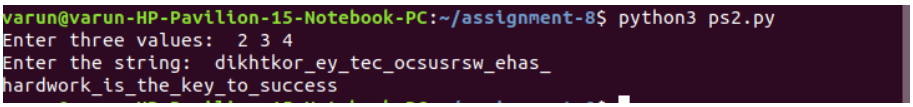
2.5 Test Cases

Input
2 3 4
dikhtkor_ey_tec_ocsussys.exit()rsw_ehas_
Output
hardwork_is_the_key_to_success

2.6 Difficulty/Issues faced

- Python donot define data type at initialization so error might crop up.
- latex is difficult to use
- Python is different then the previous languages..

2.7 Screenshots



3 Appendix

3.1 Appendix-A : code for ps1

code1

```
{
###### this is the first .py file ############
#!/usr/bin/python
import sys
import copy
######## write your code here ###########
inp = input("Enter two values:  ").split(" ")
#inp = raw_input("Enter two values:  ").split()
#give input to the
n = int(inp[0])
m = int(inp[1])
if n<=1 or n >=106:
    print("Enter proper variable")
    sys.exit()
if m<=1 or m >=106:
    print("Enter proper variable")
    sys.exit()
#a list
a=[]
str1=[]
#get the list
for i in range(n):
    #l = input("Enter the list")
    l = input("Enter the list")
    for j in range(len(l)):
        str1.append(l[j])
        str2=copy.copy(str1)
    a+=[str2]
    del str1[0:len(str1)]
#print(a)
#new matrix
Matrix1 = [[0 for x in range(m)] for y in range(n)]
for x in range(n):
    for y in range(m):
        if(a[x][y]=='S'):
            Matrix1[x][y] = 1
        else:
            Matrix1[x][y] = 0
#max2 and max1 are maximums top 2
max2=0
max1=0
mat=[]
valid=1
# to find out the cross in the matrix
```

```
for x in range(n):
    for y in range(m):
        if(Matrix1[x][y]==1):
            start=0
            valid=1
            #while valid increase valid
            while (valid==1):
                if(x>=start):
                    if(Matrix1[x-start][y]==1):
                        pass
                    else:
                        valid=0
                        break
                else:
                    valid=0
                    break

                if(y>=start):
                    if(Matrix1[x][y-start]==1):
                        pass
                    else:
                        valid=0
                        break
                else:
                    valid=0
                    break

                if(x+start<n):
                    if(Matrix1[x+start][y]==1):
                        pass
                    else:
                        valid=0
                        break
                else:
                    valid=0
                    break
                if(y+start<m):
                    if(Matrix1[x][y+start]==1):
                        pass
                    else:
                        valid=0
                        break
                else:
                    valid=0
                    break
                start=start+1
            if(max2<max1 and max1<start):
                max2=max1
            if(max1<start):
                max1=start
```

```
                mat+=[start]
                #print(start,x,y)
print(4*(max1-1)+1,4*(max2-1)+1)

}
```

Listing 1: ps1.py

3.2 Appendix-B : code for ps2

code2

```
###### this is the second .py file ###########
import sys
#rotate function
#input : list an l and length n
#output : rotate the list by  n and return the output list
def rotate(l, n):
    return l[n:] + l[:n]


#!/usr/bin/python
####### write your code here ##########
#get input
inp = input("Enter three values:  ").split(" ")
#divide it into 3 parts
k1 = int(inp[0])
k2 = int(inp[1])
k3 = int(inp[2])
if k1<1 or k1 >151:
    print("Enter proper variable")
    sys.exit()
if k2<1 or k2 >151:
    print("Enter proper variable")
    sys.exit()
if k2<1 or k2 >151:
    print("Enter proper variable")
    sys.exit()

# different parts regular expressions for matching
part1=[]
part1+='abcdefghi'        #one of the alphabets
part2=[]
part2+='jlkmnopqr'        #one of the alphabets
part3=[]
part3+='stuvwxyz_'       #one of the alphabets
#for 3 parts of the string accordint to 3 regular exp
str1=[]
str2=[]
str3=[]
#input string
string = input("Enter the string:  ")
if len(string)<1 or len(string) >151:
    print("Enter proper variable")
    sys.exit()

#convert to string
string=list(string)
#divide the string into 3 parts
```

```
for i in range(len(string)):
    if string[i] in part1:
        str1+=string[i]
        pass
    if string[i] in part2:
        str2+=string[i]
        pass
    if string[i] in part3:
        str3+=string[i]
        pass
#print(str1)
rotate(str1, k1)
#print(str2)
rotate(str2, k2)
#print(str3)
rotate(str3, k3)
#rotating the entire string according to developed lists
l=0
m=0
n=0
for i in range(len(string)):
    if string[i] in part1:
        #print(part1.index(string[i]))
        string[i]=str1[(l-k1)%len(str1)]
        l+=1
        pass
    if string[i] in part2:
        string[i]=str2[(m-k2)%len(str2)]
        m+=1
        pass
    if string[i] in part3:
        string[i]=str3[(n-k3)%len(str3)]
        n+=1
        pass
final_output = ''.join(string)
#print the final output
print(final_output)
```

Listing 2: ps2.py

References

[1] Python tutorial,
https://www.tutorialspoint.com/python/

[2] Tutorial point for Git,
https://www.tutorialspoint.com/git/

[3] Latex format guide,
https://www.sharelatex.com/