

Assignment-7

ELP - 718 Telecom Software Laboratory

Pulkit Arora

2018JTM2249

2018-2020

A report presented for the assignment on AWK AND SED



Bharti School Of
Telecommunication Technology and Management

IIT Delhi

India

August 16, 2018

Contents

1	Problem Statement-1	3
1.1	Problem Statement	3
1.2	Assumptions	3
1.3	Screen Shot	4
1.4	INPUT AND OUTPUT FORMAT	4
2	Problem Statement-2	6
2.1	Problem Statement	6
2.2	INPUT AND OUTPUT FORMAT	6
2.3	Assumptions	7
2.4	Screen Shot	7
2.5	Algorithm and Implementation	7
2.6	Code for ps1	7
2.7	Code for ps2	9

1 Problem Statement-1

1.1 Problem Statement

IT 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.

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.

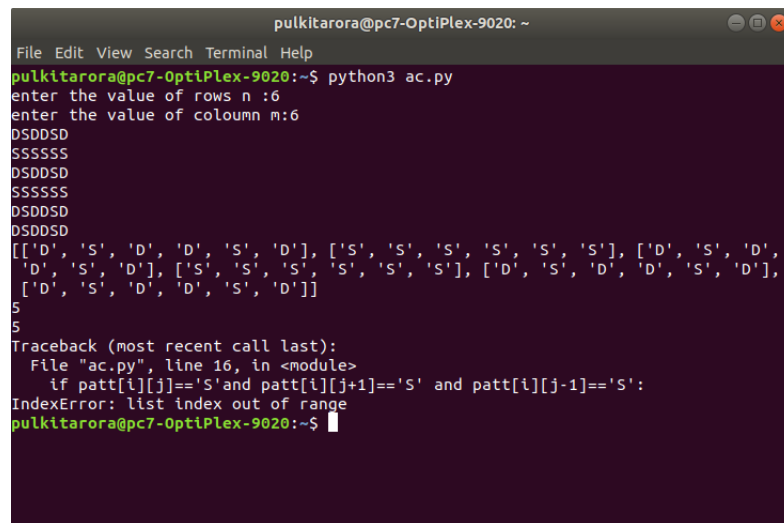
The Rate details for the plans opted by the subscribers are -

1.2 Assumptions

Constraints

- $2 \leq n \leq 105$
- $2 \leq m \leq 105$

1.3 Screen Shot



```
pulkitarora@pc7-OptiPlex-9020: ~  
File Edit View Search Terminal Help  
pulkitarora@pc7-OptiPlex-9020:~$ python3 ac.py  
enter the value of rows n :5  
enter the value of coloumn m:6  
DSDSDS  
SSSSSS  
DSDSDS  
SSSSSS  
DSDSDS  
DSDSDS  
[[('D', 'S', 'D', 'D', 'S', 'D'), ('S', 'S', 'S', 'S', 'S', 'S'), ('D', 'S', 'D',  
  'D', 'S', 'D'), ('S', 'S', 'S', 'S', 'S', 'S'), ('D', 'S', 'D', 'D', 'S', 'D'),  
  ('D', 'S', 'D', 'D', 'S', 'D')]  
5  
5  
Traceback (most recent call last):  
  File "ac.py", line 16, in <module>  
    if patt[i][j]=='S'and patt[i][j+1]=='S' and patt[i][j-1]=='S':  
IndexError: list index out of range  
pulkitarora@pc7-OptiPlex-9020:~$
```

1.4 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

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 0

```
5 6  
SSSSSS
```

SDDDS

SSSSS

SSDDSD

SSSSS

Sample Output

5 1

2 Problem Statement-2

2.1 Problem 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, k_1 , k_2 , and k_3 . 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 k_i positions in the message. Each group is rotated independently of the other two. Decrypting the message means doing a right rotation by k_i positions within each group.

2.2 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 1

```
2 3 4
dikhtkor_ey_tec_ocsusrsw_ehas_
```

Sample Output 1

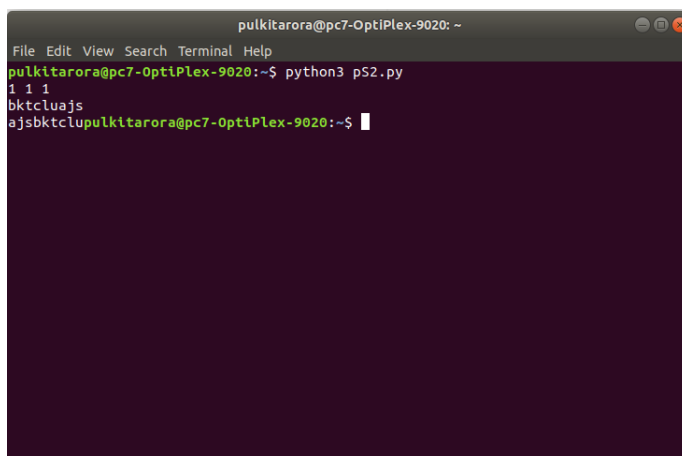
```
hardwork_is_the_key_to_success
```

2.3 Assumptions

Constraints

- $1 \leq \text{Length of the string} \leq 150$
- $1 \leq k_i \leq 150$ ($i=1,2,3$)

2.4 Screen Shot

A screenshot of a terminal window titled 'pulkitarora@pc7-OptiPlex-9020: ~'. The terminal shows the command 'python3 ps2.py' being executed. The output of the script is displayed on the next line: '1 1 1', 'bktcluajs', and 'ajsbktclupulkitarora@pc7-OptiPlex-9020:~\$'. The terminal has a dark background and a light-colored cursor.

2.5 Algorithm and Implementation

- Create a ps2.awk and output.txt file

2.6 Code for ps1

ps1.py

```
##### this is the first .py file #####  
n=int(input("enter the value of rows n:"))  
m=int(input("enter the value of coloumn m:"))  
patt = []
```

```

# loop to create matrix
for i in range(n):
    t = list(input())
    patt = patt + [t]
print(patt)

for i in range(n):
    for j in range(m):
        #print(i,j,patt[i][j])

        if patt[i][j]=='S' and patt[i][j+1]=='S' and patt[i][j-1]=='S': # loop to print a pattern of 9
            if patt[i+1][j]=='S' and patt[i+1][j-1]=='D'
            and patt[i+1][j+1]=='D':
                if patt[i+2][j-2]=='S' and patt[i+2][j+2]=='S' and patt[i+2][j+1]=='S'
                and patt[i+2][j-1]=='S' and patt[i+2][j]=='S':
                    if patt[i+3][j]=='S' and patt[i+3][j-1]=='D' and patt[i+3][j+1]=='D':
                        if patt[i+4][j]=='S'
                        and patt[i+4][j+1]=='S' and patt[i+4][j-1]=='S':
                            print("9")

        elif patt[i][j]=='S' and patt[i][j-1]=='D' and patt[i][j+1]=='D' and i<=n-1 :# loop to print a pattern of 5
            if patt[i+1][j]=='S' and patt[i+1][j+1]=='S'
            and patt[i+1][j-1]=='S':

```



```

        if patt[i+2][j]=='S' and patt[i][j-1]=='D' and patt[i][j+1]=='D':
            print("5")
    elif patt[i][j]=='S' and patt[i][j-1]=='D' and patt[i][j+1]=='D' and i<=n-1 :# loop to print a pattern of 1
        if patt[i+1][j]=='D' and patt[i+1][j-1]=='D' and patt[i+1][j+1]=='D':
            if patt[i+2][j]=='D' and patt[i+2][j-1]=='D' and patt[i+2][j+1]=='D':
                print("1")

```

write your code here

2.7 Code for ps2

ps2.py

```

##### this is the second .py file #####
#rotate
def reverse(lt,a):
    copy = list(lt)
    for i in range(len(lt)):
        if a<0:
            lt[i+a] = copy[i]
        else:
            lt[i] = copy[i-a]

#grouping

```

```
g1=" abcdefghi"  
g2=" jklmnopqr"  
g3=" stuvwxyz_"
```

```
m1 =[]  
m2 =[]  
m3 =[]  
index1=[]  
index2=[]  
index3=[]
```

```
k1,k2,k3 = list (map(int ,input().split()))
```

```
#strings  
msg = input()  
msgl = list (msg)
```

```
#compare  
for i in range(0,len(msg)):  
    if msgl[i] in g1:  
        m1.append(msgl[i])  
        index1.append(i)  
  
    elif msgl[i] in g2:  
        m2.append(msgl[i])  
        index2.append(i)  
    elif msgl[i] in g3:  
        m3.append(msgl[i])  
        index3.append(i)
```

```
#rotate c1,c2,c3
```

```
reverse(m1,k1)
```

```
reverse(m2,k2)
```

```
reverse(m3,k3)
```

```
#get decrypted
```

```
a1=0
```

```
a2=0
```

```
a3=0
```

```
for i in range(0,len(msg)+1):
```

```
    if i in index1:
```

```
        msgl[i]=m1[a1]
```

```
        a1+=1
```

```
    elif i in index2:
```

```
        msgl[i]=m2[a2]
```

```
        a2+=1
```

```
    elif i in index3:
```

```
        msgl[i]=m3[a3]
```

```
        a3+=1
```

```
for i in msgl[:]:
```

```
    print (i, end = '')
```

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

References

- [1] *The basics of python*. <http://www.linuxhowtos.org>.