Cryptography Assignment - 11

Pramod Aravind Byakod

(In Hoffstein-Pipher-Silverman book, second edition) 7.2, 7.3 (use S = 25916), 7.5 and 7.7 (you only need to find the volume)

Exercise 7.2

(a) M = (3, 7, 19, 43, 89, 195), S = 260

S>195
S-195=260-195=65
65>43
65-43=22
22>19
22-19=3
3=3
3-3=0
Solution is **[1, 0, 1, 1, 0, 1]**

(b) M = (5, 11, 25, 61, 125, 261), S = 408

S>261
S-261=408-261=147
147>125
147-125=22
22>11
22-11=11
11>5
11-5=6
Solution doesn't exist in this case

(c) M = (2, 5, 12, 28, 60, 131, 257), S = 334

S>257
S-257=334-257=77
77>60
77-60=17
17>12
17-12=5
5=5
Solution is **[0, 1, 1, 0, 1, 0, 1]**

(d) M = (4, 12, 15, 36, 75, 162), S = 214

S=214>162
S-162=214-162=52
52>36
52-36=16
16>15
16-15=1
1<4(the smallest value of M)
Solution doesn't exist in this case

Exercise 7.5

(a)

```
B = matrix([[1, 3, 2],[2, -1, 3],[1, 0, 2]])
Bt = matrix([[-1, 0, 2],[3, 1, -1],[1, 0, 1]])
result = Bt*B.inverse()
print ("The change of basis matrix that transforms B' into B is:\n" +str(result.inverse()))
```

Output:

```
The change of basis matrix that transforms B' into B is:
[ 13/3     3 -11/3]
[   -1    -1     4]
[  1/3     0   4/3]
```

(b)

```
v=vector([2,3,1])
w=vector([-1,4,-2])
len_v=sqrt(2^2+3^2+1^2)
len_w=sqrt((-1)^2+4^2+(-2)^2)
dot_product=v*w
cos_angle=dot_product/(len_v*len_w)
angle = (180*arccos(cos_angle)/pi).n()
print ("Length of v is "+str(len_v.n()))
print ("Length of w is "+str(len_w.n()))
print ("Dot product of v and w is "+str(dot_product))
print ("Angle between v and w is "+str(angle)+" degree")
```

Output:

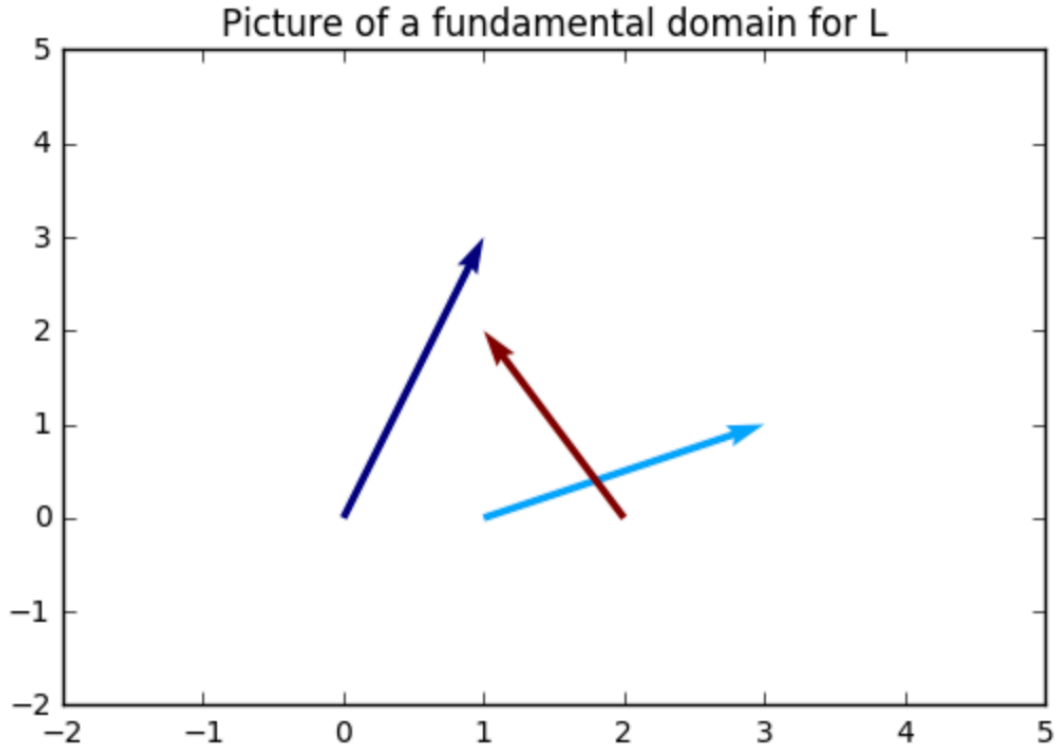
```
Length of v is 3.74165738677394
Length of w is 4.58257569495584
Dot product of v and w is 8
Angle between v and w is 62.1881568617839 degree
```

Exercise 7.7

```
A = matrix([[1, 3, -2],[2, 1, 0],[-1, 2, 5]])
print("Output:")
print 'The volume of fundamental domain is '+ str(abs(det(A)))
```

```
Output:
The volume of fundamental domain is 35
```

```
import numpy as np
import matplotlib.pyplot as plt
A = np.array([[1, 3, -2],[2, 1, 0],[-1, 2, 5]])
X, Y, U = zip(*A)
ax = plt.gca()
ax.quiver(X, Y, U, angles='xy', scale_units='xy', scale=1)
ax.set_xlim([-2, 5])
ax.set_ylim([-2, 5])
plt.title("Picture of a fundamental domain for L")
plt.draw()
plt.show()
```



Exercise 7.3
M = (5186, 2779, 5955, 2307, 6599, 6771, 6296, 7306, 4115, 637)
S = 25916
A = 4392
B = 8387

Using Sage, xgcd(4392, 8387) = (1, 2683, -1405)
We can write 1=2683*4392-1405*8387
Therefore, inverse of 4392 is 2683

```
A=4392
B=8387
S=25916
Inv_A=xgcd(A, B)[1]
print ("Inverse of A is "+str(Inv_A))
R = Integers(B)
M=(5186, 2779, 5955, 2307, 6599, 6771, 6296, 7306, 4115, 637)
print ("Private sequence r is: ")
for i in M:
    print R(i*Inv_A),
Sp=R(Inv_A*S)
print (" \nDisguised S is "+str(Sp))
```

```
Inverse of A is 2683
Private sequence r is:
5 14 30 75 160 351 750 1579 3253 6510
Disguised S is 4398
```

Decrypt the message:
Sp=4398
4398>3253
4398-3253=1145
1145>750
1145-750=395
395>351
395-351=44
44>30
44-30=14
14=14
So, the result is **[0, 1, 1, 0, 0, 1, 1, 0, 1, 0]**