HW10ImplementationSolutions

December 7, 2021

[1]:
```
########## Preamble:
def fastPowerSmall(g,A,N):
    a = g
    b = 1
    while A>0:
        if A % 2 == 1:
            b = b * a % N
        A = A//2
        a = a*a % N
    return b

def getBinary(A):
    binaryList = []
    while A>0:
        if A%2 == 0:
            binaryList.append(0)
        else:
            binaryList.append(1)
        A = math.floor(A/2)
    return binaryList

def extendedEuclideanAlgorithm(a,b):
    u = 1
    g = a
    x = 0
    y = b
    while true:
        if y == 0:
            v = (g-a*u)/b
            return [g,u,v]
        t = g%y
        q = (g-t)/y
        s = u-q*x
        u = x
        g = y
        x = s
        y = t
```

```
def findInverse(a,p):
    inverse = extendedEuclideanAlgorithm(a,p)[1] % p
    return inverse
```

[2]:
```
##########Problem 1
#####Part (a)
#This just checks that the discriminant is nonzero
def isCurve(E,p=0):
    A,B = E
    Delta = 4*A**3 + 27*B**2
    if p!=0:
        Delta = Delta % p
    if Delta!=0:
        return True
    else:
        return False

#####Part (b)
#This just checks if the point is on the curve
def onCurve(E,P,p=0):
    if P=='O':
        return True
    A,B = E
    x,y = P
    LHS = y**2
    RHS = x**3 + A*x + B
    if p!=0:
        LHS = LHS % p
        RHS = RHS % p
    if LHS==RHS:
        return True
    else:
        return False

#####Part (c)
primeList = [3,5,7,11,13,17,19]
E = [3,2]
P = [3,5]
for p in primeList:
    print("E a curve over",p,":",isCurve(E,p))
    if(isCurve(E,p)):
        print("P is on E over",p,":",onCurve(E,P,p))
        print("O is on E over",p,":",onCurve(E,'O',p))

#####Part (d)
pointList = ['O']
```

```
for i in range(0,6):
    for j in range(0,6):
        if onCurve(E,[i,j],7):
            pointList.append([i,j])
print(pointList)
```

```
E a curve over 3 : False
E a curve over 5 : True
P is on E over 5 : False
O is on E over 5 : True
E a curve over 7 : True
P is on E over 7 : False
O is on E over 7 : True
E a curve over 11 : True
P is on E over 11 : False
O is on E over 11 : True
E a curve over 13 : True
P is on E over 13 : True
O is on E over 13 : True
E a curve over 17 : True
P is on E over 17 : False
O is on E over 17 : True
E a curve over 19 : True
P is on E over 19 : False
O is on E over 19 : True
['O', [0, 3], [0, 4], [2, 3], [2, 4], [4, 1], [5, 3], [5, 4]]
```

[3]:
```
##########Problem 2

#####Part (a)
def addPoints(E,P,Q,p):
    #First see if you're adding O
    if P=='O':
        return Q
    if Q=='O':
        return P
    #Otherwise let's extract some data
    A,B = E
    x1,y1 = P
    x2,y2 = Q
    #make sure everything is reduced mod p
    x1 = (x1 % p)
    x2 = (x2 % p)
    y1 = (y1 % p)
    y2 = (y2 % p)

    #If the points are inverses we just return the point at infinity
```

```
    if y1!=y2 and x1==x2:
        return 'O'

    #Otherwise we begin by computing the slope of the line
    if(x1==x2):
        L = ((3*x1**2 + A)*findInverse(2*y1,p)) % p
    else:
        L = ((y2-y1)*findInverse(x2-x1,p)) % p

    #Finally compute coords of the new points
    x3 = (L**2 - x1 - x2) % p
    y3 = (L*(x1-x3) - y1) % p
    return [x3,y3]

#####Part (b)
print("Curve: y^2 = x^3 + 3x + 8 over F_{13}")
print("P = (9,7) and Q= (1,8)")
E = [3,8]
p = 13
P = [9,7]
Q = [1,8]
print("P+Q=",addPoints(E,P,Q,p))
print("2P=",addPoints(E,P,P,p))
print("O+Q=",addPoints(E,'O',Q,p))
print("")
print("Curve: y^2 = x^3 + 3x + 2 over F_7: Multiplication Table")
print("")
for P in pointList:
    for Q in pointList:
        R = addPoints(E,P,Q,7)
        if R=='O':
            print("[OOOO]",end='')
        else:
            print(R,end=''),
    print("")
print("")
print("Curve: y^2 = x^3 + 231x + 473 over F_{17389}")
print("P =  (11259, 11278) and Q = (11017,14673)")
E = [231,473]
p = 17389
P = [11259,11278]
Q = [11017,14673]
print("P+Q=",addPoints(E,P,Q,p))
print("2Q=",addPoints(E,Q,Q,p))
print("3P=",addPoints(E,P,addPoints(E,P,P,p),p))
print("")
```

```
Curve: y^2 = x^3 + 3x + 8 over F_{13}
P = (9,7) and Q= (1,8)
P+Q= [2, 10]
2P= [9, 6]
O+Q= [1, 8]

Curve: y^2 = x^3 + 3x + 2 over F_7: Multiplication Table

[OOOO][0, 3][0, 4][2, 3][2, 4][4, 1][5, 3][5, 4]
[0, 3][2, 3][OOOO][5, 4][0, 4][5, 3][2, 4][4, 6]
[0, 4][OOOO][2, 4][0, 3][5, 3][4, 6][4, 1][2, 3]
[2, 3][5, 4][0, 3][4, 6][OOOO][2, 4][0, 4][4, 1]
[2, 4][0, 4][5, 3][OOOO][4, 1][5, 4][4, 6][0, 3]
[4, 1][5, 3][4, 6][2, 4][5, 4][0, 3][2, 3][0, 4]
[5, 3][2, 4][4, 1][0, 4][4, 6][2, 3][5, 4][OOOO]
[5, 4][4, 6][2, 3][4, 1][0, 3][0, 4][OOOO][5, 3]

Curve: y^2 = x^3 + 231x + 473 over F_{17389}
P =  (11259, 11278) and Q = (11017,14673)
P+Q= [12613, 2831]
2Q= [522, 6187]
3P= [13395, 14468]
```

[0]: