Homework9

November 23, 2020

[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
```

[37]:
```
##########Problem 1

def PollardRhoLog(g,h,p,n=-1):
    #If we don't know n we assume it is p-1
    if n==-1:
        n = 2*isqrt(p-1)

    #We should also reduce g and h mod p
    g = g % p
    h = h % p
    #We need both a list and a set in order to remember the logarithm
    list1 = []
    list1set = set()

    #Add a bunch of random powers of g to your lists
    for r in range(0,n):
        i = ZZ.random_element(0,p)
        x = fastPowerSmall(g,i,p)
        #In list 1 we also save the exponent
        list1.append([x,i])

        #We also save to a set for faster searching
        list1set.add(x)

    #Compute a bunch of random hg^j's to try and get a collision.
    for r in range(0,n):
```

```
        j = ZZ.random_element(0,p)
        x = (h*fastPowerSmall(g,j,p)) % p

        #See if your thing is in list1
        if x in list1set:
            #If we're in the set find the index!
            #Notice we only have to do this once!
            for l in range(0,n):
                if x == list1[l][0]:
                    #We found the match!
                    i = list1[l][1]
                    #Since g^i = hg^j the discrete log is i-j
                    return (i-j) % (p-1)

    #If we got here then there was no match
    print("No overlap, either h is not a power of g or you should increase the␣
↪size of your lists")
    return -1
```

[38]:
```
##########Problem 2
#####Part (a)
g = 2
h = 390
p = 659
log = PollardRhoLog(g,h,p)
print("We computed the log to be",log)
power = fastPowerSmall(g,log,p)
print("As a check we raise",g,"to the",log,"and get",power)
print("Does this match?",h)

####Part (b)
g= 10
h = 106
p = 811
log = PollardRhoLog(g,h,p)
print("We computed the log to be",log)
power = fastPowerSmall(g,log,p)
print("As a check we raise",g,"to the",log,"and get",power)
print("Does this match?",h)
```

```
collision! 427
We computed the log to be 177
As a check we raise 2 to the 177 and get 390
Does this match? 390
collision! 656
We computed the log to be 645
As a check we raise 10 to the 645 and get 106
```

Does this match? 106

[0]: