Project8: impl sm2 with RFC6979

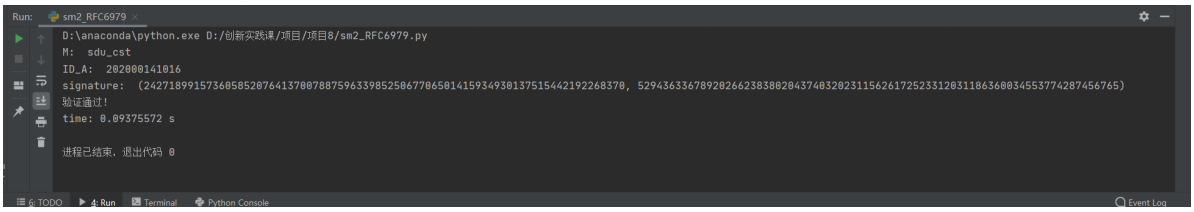
随机生成k用于签名

代码

```
# 椭圆曲线上的点乘   将num展为二进制判断每一位为0或1进行计算会很快的得到结果
def funcmult(num, P,a,p):
    #print("funcmult")

    num = bin(num)[2:]
    qx, qy = P[0], P[1]
    Q = [qx, qy]
    for i in range(1, len(num)):
        Q = funcadd(Q, Q, a, p)
        if num[i] == '1':
            Q = funcadd(Q, P, a, p)
    return  Q

def sign(M,ID_A,G,PA,dA,a,b,p):  # M为待签名的消息

    ZA = precompute(ID_A,a,b,G,PA[0],PA[1])
    #print("1")
    M_ = ZA + M
    str_m = bytes(M_,encoding='utf-8')
    e = sm3.sm3_hash(func.bytes_to_list(str_m))
    #print("2")
    r = 0
    k1 = random.randrange(1, n - 1)  #随机生成

    s = 0
    while (r==0 or ((r+k1)%n ==0) or s==0):

        k1 = random.randrange(1,n-1)
        kG = funcmult(k1,G,a,p)

        r = (int(e,16)+kG[0]) % n

        s = dn*(k1-r*dA)%n
    #print("3")
    return r,s


def verify(ID_A,M,r,s,PA,a,b,p):

    if r<1 or r>(n-1) or s<1 or s>(n-1):
        print("False!")
        return False

    ZA = precompute(ID_A, a, b, G, PA[0], PA[1])
    M_ = ZA + M
    #print("$")
    str_m = bytes(M_, encoding='utf-8')
```

```
    e = sm3.sm3_hash(func.bytes_to_list(str_m))
    #print("%")
    t = (r + s)% n
    tem1 = funcmult(s,G,a,p)
    tem2 = funcmult(t,PA,a,p)
    point = funcadd(tem1,tem2,a,p)
    #print("7")
    R = (int(e,16) + point[0])%n

    if R==r:
        print("验证通过!")
        return True
```

运行结果

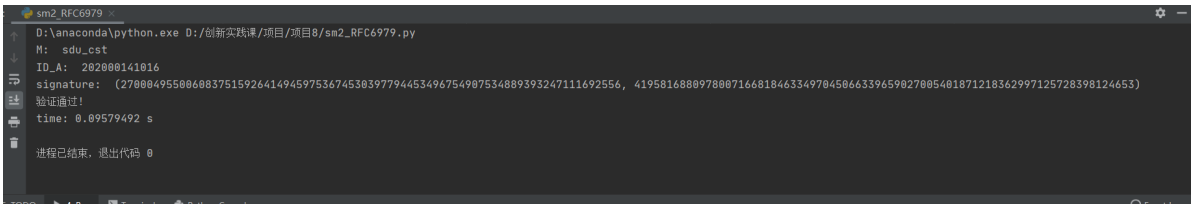


RFC6979生成k

代码

```
def sign(M,ID_A,G,PA,dA,a,b,p):  # M为待签名的消息

    ZA = precompute(ID_A,a,b,G,PA[0],PA[1])
    #print("1")
    M_ = ZA + M
    str_m = bytes(M_,encoding='utf-8')
    e = sm3.sm3_hash(func.bytes_to_list(str_m))
    #print("2")
    r = 0

    #将M||ID_A||PA作为sm3的输入得到哈希值作为最终的k
    #依次减少不同的人用了相同的随机数k的概率
    str_k = M + ID_A + hex(PA[0])[2:] + hex(PA[1])[2:]
    str_k = bytes(str_k,encoding='utf-8')
    str_k = sm3.sm3_hash(func.bytes_to_list(str_k))
    k1 = int(str_k,16)
    s = 0
    while (r==0 or ((r+k1)%n ==0) or s==0):

        k1 = random.randrange(1,n-1)
        kG = funcmult(k1,G,a,p)

        r = (int(e,16)+kG[0]) % n

        s = dn*(k1-r*dA)%n
    #print("3")
    return r,s
```

运行结果


sm2_RFC6979
D:\anaconda\python.exe D:/创新实践课/项目/项目8/sm2_RFC6979.py
M: sdu_cst
ID_A: 202000141016
signature: (27000495500608375159264149459753674530397794453496754907534889393247111692556, 41958168809780071668184633497045066339659027005401871218362997125728398124653)
验证通过!
time: 0.09579492 s
进程已结束，退出代码 0