RSA算法的实现方法Java版放送 PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/245/2021_2022_RSA_E7_AE_97_E6_B3_95_E7_c104_245774.htm 一开始不知道有BigInteger这个大数类，居然自己去实现了一个，写了大数加法后，才发现有现成的T_T以下是引用片段：package rsa. import java.math.BigInteger. public class RSA { private long p,q,e,d,n. public RSA(){ int pIndex = (int)(Math.random()*10). int qIndex. int eIndex. do{ qIndex = (int)(Math.random()*10). } while(qIndex==pIndex). do{ eIndex = (int)(Math.random()*10). } while(eIndex==pIndex||eIndex==pIndex). p = 1033. q = 2017. e = 29437. n = p*q. d = calculateD(). } private long calculateD(){ long t0 = 0,t1 = 1,t2 = -1. long r0 = (p-1)*(q-1), m = r0,r1 = e ,r2 = -1. do{ long q = r0/r1. r2 = r0-r1*q. if(r2==0)break. t2 = t0 - t1*q. while(t2 t2 =m. } if(t2>=m){ t2 %= m. } r0 = r1. r1 = r2. t0 = t1. t1 = t2. }while(r2!=0). if(r1!=1){ return 0. } else{ return t2. } } public long getE() { return e. } public long getN() { return n. } public long getD() { return d. } public BigInteger encode(BigInteger data){ return pow(data,d).mod(new BigInteger(n "")). } public BigInteger decode(BigInteger code){ return pow(code,e).mod(new BigInteger(n "")). } public BigInteger pow(BigInteger data,long p){ data = data.pow((int)p). return data. } public static void main(String args[]){ RSA rsa = new RSA(). BigInteger data = new BigInteger("222222"). long oldtime = System.currentTimeMillis(). BigInteger code = rsa.encode(data). long newtime = System.currentTimeMillis(). double codetime =

((double)(newtime-oldtime))/1000. oldtime = System.currentTimeMillis(). BigInteger decode = rsa.decode(code). newtime = System.currentTimeMillis(). double decodetime = ((double)(newtime-oldtime))/1000. System.out.println("privateKey:" rsa.d). System.out.println("publickKey:" rsa.e). System.out.println("N:" rsa.n). System.out.println("data:" data). System.out.println("code:" code " time:" codetime). System.out.println("decode:" decode " time:" decodetime). } }