On my honor, as a University of Colorado at Boulder student, I have neither given nor received unauthorized assistance.

1. If x is m-bits long and y is n-bits long, the run time of the function is O(m\*n). The function is called recursively n times (y/2 performs a right shift, so n bits requires n shifts to complete) and the addition operation x + 2z requires one operation for every bit in x. In summary, there are n recursive calls, each requiring m operations, for a total of O(m\*n).

2. gcd(770,546)

a. 770 = 2 \* 5 \* 7 \* 11

546 = 2 \* 3 \* 7 \* 13

gcd(770,546) = 2 \* 7 = 14

b. gcd(a,b) = gcd(b, a mod b)

gcd(770,546) = gcd(546,224) = gcd(224,98) = gcd(98,28) = gcd(28,14) = gcd(14,0)

gcd(770,546) = 14

c. eE(a,b): //(extendedEuchlid)

x',y',d = eE( b, a mod b)

return y', x' - floor(a/b)\*y', d

|  |  |  |
| --- | --- | --- |
| Iteration | x', y' d | return value |
| eE(770,546) | 7,-17,14 | -17,24,14 |
| eE(546,224) | -3,7,14 | 7,-17,14 |
| eE(224,98) | 1,-3,14 | -3,7,14 |
| eE(98,28) | 0,1,14 | 1,-3,14 |
| eE(28,14) | 1,0,14 | 0,1,14 |
| eE(14,0) |  | 1,0,14 |

extendedEuclid(770,546) = -17,24,14

**a = 770, b = 546, d = 14, x = -17, y = 24**

-17\*770 + 24\*546 = 14; gcd(770,546) = 14

3. 77293 mod 342 ≡ (73)2431 mod 342 ≡ (343)2431 mod 342 ≡ 12431 mod 342 ≡ 1

4. Times from three runs of RSA encryption/decryption with different key lengths:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Time to find keys  (Including p, q, N, phi, e, and d) | Time to Encrypt Message | Time to Decrypt Message |
| 8 bit key | 0.0002529621124 | 2.1457672e-06 | 2.8610229e-06 |
| 16 bit key | 0.0008039474487 | 4.0531158e-06 | 3.0994415e-06 |
| 24 bit key | 0.0021779537200 | 3.8146972e-06 | 1.5974044e-05 |