Quant. Comp. HW - 2

Steven MacCoun

Oct. 18, 2005

1 Simon's Problem

2 Modular Exponentiation

Here was my python code:

```
#Modular Exponentiation
import math

def modular_exponentiation(base, exponent, modulus):
         c = 1
         for e_prime in range(1, exponent+1):
                    c = (c * base) % modulus
        return c

print modular_exponentiation(1234, 1234*1234, math.pow(10, 10))
```

And the output was:

$$\boxed{3102217216.0}$$

3 RSA Misuse

4 Prime factorization

Problem: Consider n=121932632103337941464563328643500519

(a) How many bits is n?

```
print len(str(121932632103337941464563328643500519))
```

Output:

$$\boxed{36}$$

(b) Find if n is prime with program that runs in less than one second.

```
def miller_rabin_pass(a, s, d, n):
         a_to_power = pow(a, d, n)
         if a_to_power == 1:
                   return True
         for i in xrange(s-1):
                did if a_to_power == n - 1:
```

```
def miller_rabin_pass(a, s, d, n):
         a_to_power = pow(a, d, n)
         if a_to_power == 1:
                   return True
         for i in xrange(s-1):
                if a_to_power == n - 1:
                          return True
                a_to_power = (a_to_power * a_to_power) % n
        return a_to_power == n - 1

def miller_rabin(n):
#compute s and d
d = n - 1
s = 0
while d % 2 == 0:
d >>= 1
s += 1

#Run several miller_rabin passes
for repeat in xrange(20):
a = randint(2, n-1)
if not miller_rabin_pass(a, s, d, n):
return False
return True

print miller_rabin(n)
```