Math 4330, Homework 6, 2/24/2014 : Due 3/10/2014 [1]

Read 4.5.4 up to and including page 386.

1. Suppose N is an odd composite number divisible by at least two distinct primes and $S = \{(x, y) \in \mathbb{Z}^2 \ : \ x^2 \equiv y^2 \pmod{\text{N}}\}$.
   (i) Prove that if (u, v) is chosen randomly from S, there is at least a 50% chance that $u \not\equiv \pm v \pmod{\text{N}}$.
   (ii) If $(u, v) \in S$ and $u \not\equiv \pm v \pmod{\text{N}}$, explain how these two integers can be used to find a proper divisor of N.

2. Write a function `trialDivision(n)` to find a proper divisor of each of the following numbers: $3^{15} + 40$, $3^{19} + 22$, $3^{25} + 16$, $3^{29} + 8$, $3^{31} + 26$, $5^{25} + 6$. Your program should also report the time required to factor each number, using the following code snippet:

```
import datetime
...
start = datetime.datetime.now()
# Code to be timed here.
...
stop = datetime.datetime.now()
elapsed = stop - start
elapsedSeconds = elapsed.seconds + elapsed.microseconds/1000000.0
print "Elapsed time %f seconds." % (elapsedSeconds)
```

3. Use Algorithm B (omit Step B2 and include the suggestion on p. 386 to handle failures) to find divisors of the same numbers from the previous exercise (it need not completely factor the numbers - just find a single proper divisor of each). Your program should also report the time required to factor each number. Compare the timing results from this and the previous exercise side-by-side in a table.

4. Among all composite numbers of the form $N = pq$ with p and q prime and $\sqrt{N}/2 \leq p \leq 2\sqrt{N}$, roughly what is the largest such number that each of your programs could compute if they ran for one year nonstop? You should turn in a written solution with experimental results and a carefully reasoned argument supporting your conclusion.