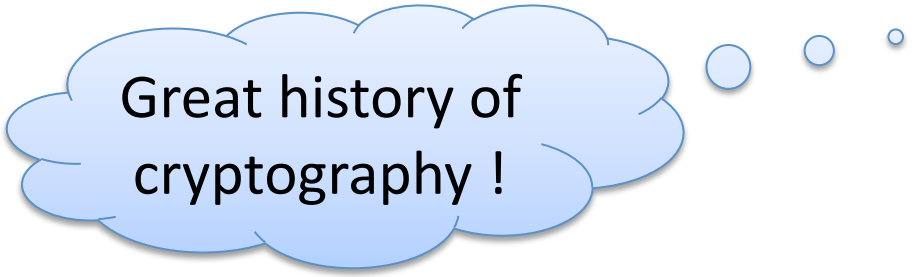
CSE 321 Discrete Structures

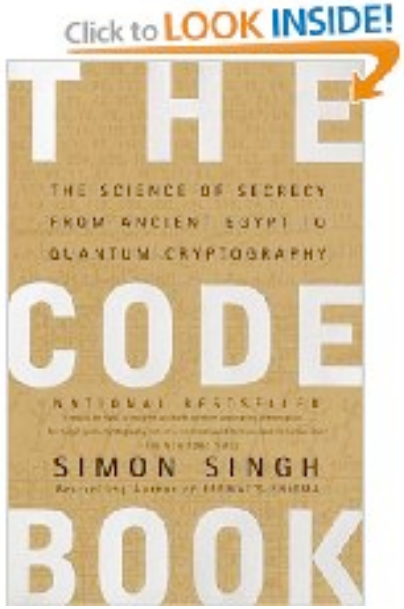
February 12th, 2010

Lecture 16: Prime Numbers

Highlights from Lecture 15

- Fermat's theorem
- Public Key Cryptography





Distribution of Primes

2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71
73 79 83 89 97 101 103 107 109 113 127 131 137 139 149
151 157 163 167 173 179 181 191 193 197 199 211 223
227 229 233 239 241 251 257 263 269 271 277 281 283
293 307 311 313 317 331 337 347 349 353 359

- We search for large prime numbers
- Are they easy to find ?
- How many prime numbers are there ???

Distribution of Primes

Bertrand-Chebyshev theorem:

> For any $n > 0$ there exists at least one prime number p such that $n < p < 2n$

- What does this tell us in practice ?

Prime Number Theorem

- For every number n, let

$$\pi(n) = |\{ p \mid p \text{ is prime and } p \leq n \}|$$

2 **3** 4 **5** 6 **7** 8 9 10 **11** 12 **13** 14 15 …

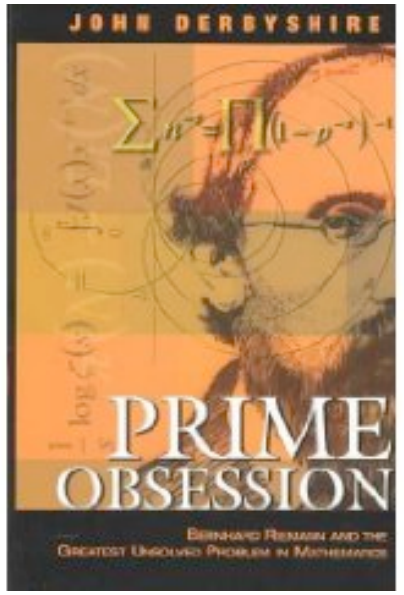
$$\pi(10) = 4$$
$$\pi(11) = 5$$
$$\pi(12) = 5$$
$$\pi(13) = 5$$
$$\pi(14) = 6$$
. . . . .

Prime Number Theorem

THEOREM $\lim_{n \to \infty} \pi(n) / (n / \ln(n)) = 1$

What does this tell us in practice ?

Has fascinating history



Famous Algorithmic Problems

- Primality Testing:
  - Given an integer n, determine if n is prime
  - Efficient algorithms exists (see next)

- Factoring
  - Given an integer n, determine the prime factorization of n
  - It is believed (HOPED !!!!) that no efficient algorithms exists

Primality Testing

- Is the following 200 digit number prime:

4099240841609602817976123253258752540290928509908622013340392052540955208352860621543991594826087571889379782473511862113819256949084009806113306665025560806560925390128880130203544188487818794421903 3

Showing a number is NOT prime

- Trial division by small primes
- Fermat's little theorem
  - $a^{p-1} \bmod p = 1$ if p is prime
- Miller's Test
  - if p is prime, the only square roots of one are 1 and -1
  - if p is composite other numbers can be the square root of one
  - repeated squaring used to find a non-trivial square root of one from a starting value b

Probabilistic Primality Testing

- Conduct Miller's test for a random b
  - If p is prime, it always passes the test
  - If p is not prime, it fails with probability ¾
- Primality testing
  - Choose 100 random b's and perform Miller's test on each
  - If any say false, answer "Composite"
  - If all say true, answer "Prime"

Exponentiation

- Compute 78365^{65536}
- Compute $78365^{65536} \bmod 104729$
- Compute $78365^{81453} \bmod 104729$

$$81453 = 2^{16} + 2^{13} + 2^{12} + 2^{11} + 2^{10} + 2^{9} + 2^{5} + 2^{3} + 2^{2} + 2^{0}$$

Fast exponentiation

```
/* pre condition: a ≥ 0 */
{ int x = a; int y = b; int z = 0;
  { while (x > 0) {
     /* loop invariant:   b^a = y^x * z ∧ x ≥0  */
     if ((x & 1) == 0) { x >>= 1; y = y*y; }
     else { x--;     z *= y; }
  }
/* post condition: z = b^a */
```

Why “fast” ?