

MA/CSSE 473

Day 10

Primality Testing



MA/CSSE 473 Day 10

- HW 4 is due now. HW 5 is available on-line
- HW 6 includes an implementation problem (described in the HW5 document)
 - You can work with another student
 - Start soon
- Exam 1: Tuesday, September 30
 - You may bring your textbook, plus a one-sided 8.5x11 inch piece of paper containing anything that you can read unaided or with normal eyeglasses
- **Student Questions**
- Textbook topics
- Fermat's Little Theorem proof
- Primality Testing



Textbook Topics I Won't Cover

- **Chapter 1 topics** that I will not discuss in detail unless you have questions. They should be review:
 - Sieve of Eratosthenes
 - Algorithm Specification, Design, Proof, Coding
 - Problem types : sorting, searching, string processing, graph problems, combinatorial problems, geometric problems, numerical problems
 - Data Structures: ArrayLists, LinkedLists, Graphs and their representations, Weighted graphs (a.k.a Networks), trees, search trees, sets, dictionaries,



Textbook Topics I Won't Cover

- Chapter 2
 - Empirical analysis of algorithms should be review
 - I believe that we have covered everything else in the chapter except amortized algorithms and recurrence relations
 - We will discuss amortized algorithms
 - Recurrence relations are covered in CSSE 230 and MA 375. We'll review particular types as we encounter them.



Textbook Topics I Won't Cover

- Chapter 3 - Review
 - Bubble sort, selection sort, and their analysis
 - Sequential search and string matching
- Probably new, but quite simple to understand
 - Closest Pair and Convex Hull
 - Traveling Salesman
 - Knapsack
 - Assignment



Textbook Topics I Won't Cover

- Chapter 4 - Review
 - Mergesort, quicksort, and their analysis
 - Binary search
 - Binary Tree Traversals



Textbook Topics I Won't Cover

- Chapter 5 - Review
 - Insertion Sort and its analysis
 - Depth-first search and Breadth-first Search
 - Binary Tree Traversals
 - Interpolation Search
 - Search, insertion, delete in Binary Tree
 - AVL tree insertion and rebalance



Recap: Fermat's Little Theorem

- **Formulation 1:** If p is prime, then for every number a with $1 \leq a < p$, $a^{p-1} \equiv 1 \pmod{p}$
- **Formulation 2:** If p is prime, then for every number a with $1 \leq a < p$, $a^p \equiv a \pmod{p}$
- These are clearly equivalent.
- We will examine a combinatorial proof of the first formulation.



Fermat's Little Theorem: Proof (part 1)

- **Formulation 1:** If p is prime, then for every number a with $1 \leq a < p$, $a^{p-1} \equiv 1 \pmod{p}$
- Let $S = \{1, 2, \dots, p-1\}$
- **Lemma**
 - Multiplying all of the numbers in S by $a \pmod{p}$ permutes S
- **Example:** $a=3$, $p=7$
 - $1 \rightarrow 3 \rightarrow 2 \rightarrow 6 \rightarrow 4 \rightarrow 5 \rightarrow 1$
- **Proof of the lemma**
 - Suppose that $a \cdot i \equiv a \cdot j \pmod{p}$.
 - Since p is prime and $a \neq 0$, a has an inverse.
 - Multiplying both sides by a^{-1} yields $i \equiv j \pmod{p}$.
 - Thus, multiplying the elements of S by $a \pmod{p}$ takes each element to a different element of S .
 - Thus (by the pigeonhole principle), every number $1..p-1$ is $a \cdot i \pmod{p}$ for some i in S .



Fermat's Little Theorem: Proof (part 2)

- **Formulation 1:** If p is prime, then for every number a with
 $1 \leq a < p$, $a^{p-1} \equiv 1 \pmod{p}$
- Let $S = \{1, 2, \dots, p-1\}$
- **Lemma (which we proved last time)**
 - Multiplying all of the numbers in S by $a \pmod{p}$ permutes S
- **Therefore:**
 $\{1, 2, \dots, p-1\} = \{a \cdot 1 \pmod{p}, a \cdot 2 \pmod{p}, \dots, a \cdot (p-1) \pmod{p}\}$
- Take the product of all of the elements on each side .
 $(p-1)! \equiv a^{p-1}(p-1)! \pmod{p}$
- Since p is prime, $(p-1)!$ is relatively prime to p , so we can divide both sides by it to get the desired result.



Easy Primality Test

- Is N prime?
- Pick some a with $1 < a < N$
- Is $a^{N-1} \equiv 1 \pmod{N}$?
- If so, N is prime; if not, N is composite
- But, wait a minute!
 - Fermat's Little Theorem is not an "if and only if" condition.
 - It doesn't say what happens when N is not prime.
 - **Example:** 341 is not prime (it is $11 \cdot 31$), but $2^{340} \equiv 1 \pmod{341}$
- We can hope that if N is composite, the test will fail for most values of a .
- It turns out that this hope is well-founded.



A Better Primality Test?

- Is N prime?
 - Pick a random positive integer, a , from $1, \dots, N-1$
 - Is $a^{N-1} \equiv 1 \pmod{N}$?
 - If so, N is prime; if not, N is composite
- Does this work?
- Of course it doesn't work!
 - N may not be prime, but we might just happen to pick an a for which $a^{N-1} \equiv 1 \pmod{N}$
- Carmichael numbers (extremely rare):
 - Composite, but $a^{N-1} \equiv 1 \pmod{N}$ for all a that are relatively prime to N
 - The smallest Carmichael number is 561
 - We'll see later how to deal with those
 - How rare are they?

n	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
$C(10^n)$	1	7	16	43	105	255	646	1547	3605	8241	19279	44706	105212	246683	585355	1401644	3381806	8220777

- For now, we assume a Carmichael-free world



How many "false positives"?

- If N is composite, and $1 \leq a < N$, how likely is it that a^{N-1} is not congruent to $1 \pmod{n}$?
- If a^{N-1} is not congruent to $1 \pmod{n}$ for some a that is relatively prime to N , then it must be true for at least half of the choices of $a < N$.
 - Let b be some number (if any exist) such that it passes the Fermat test, i.e. $b^{N-1} \equiv 1 \pmod{N}$.
 - Then the number $a \cdot b$ fails the test:
 - $(ab)^{N-1} \equiv a^{N-1}b^{N-1} \equiv a^{N-1}$, not congruent to $1 \pmod{N}$.
 - Diagram on whiteboard.
 - For a fixed a , $f(b) = ab$ is a one-to-one function, so there are at least as many b 's that fail the Fermat test as pass it.



Where are we now?

- We ignore Carmichael numbers for now.
- If N is prime, $a^{N-1} \equiv 1 \pmod{N}$ for all $0 < a < N$
- If N is not prime, then $a^{N-1} \equiv 1 \pmod{N}$ for at most half of the values of $a < N$.
- $\Pr(\text{algorithm returns } \mathbf{true} \text{ if } N \text{ is prime}) = 1$
 $\Pr(\text{algorithm returns } \mathbf{true} \text{ if } N \text{ is composite}) \leq \frac{1}{2}$
- How to reduce the probability of error?



The algorithm

- To test N for primality
 - Pick positive integers $a_1, a_2, \dots, a_k < N$ at random
 - if $a_i^{N-1} \equiv 1 \pmod{N}$ for all $i = 1, 2, \dots, k$:
 - return the result of the Carmichael # test
 - else:
 - return false



Carmichael Numbers

- A Carmichael N number is a composite number that passes the Fermat test for all a with $0 < a < N$
- A way around them (Rabin and Miller). For some t and u , $N-1 = 2^t u$.
- As before, compute $a^{N-1} \pmod{N}$, but do it this way:
 - Calculate $a^u \pmod{N}$, then repeatedly square, to get the sequence
 $a^u \pmod{N}, a^{2u} \pmod{N}, \dots, a^{2^t u} \pmod{N} \equiv a^{N-1} \pmod{N}$
- More next time!



Generating Random Primes



Amortized algorithms

- P49-50
- Growable array exercise from 220/230

