CS5800 – ALGORITHMS

MODULE 2. COMPLEXITY (AND COMPUTABILITY & CRYPTOGRAPHY)

Lesson 2: GCD

Ravi Sundaram

Topics

- GCD
- Primes & Composites
- Fundamental Theorem of Arithmetic
- Factoring
- Euclid’s Algorithm
  - Correctness
  - Complexity
- Summary

GCD

- GCD(A,B) = largest number D such that D is a divisor of A and D is a divisor of B

- Examples:
  - GCD(12, 30) = 6
  - GCD(2257, 3337) = ?

- Why do we care about GCD?
  - Basic operation in modular arithmetic
  - Crucial for cryptography

Primes & Composites

- Prime – number that is divisible only by 1 and itself
  - 2, 3, 5, 7, 11, 13, …
  - Composite – opposite of prime (1 is neither prime nor composite)
  - 4, 6, 8, 9, 10, …
- How many primes are there?
  - Infinite number
- Why are primes special?
  - primes:numbers :: atoms:molecules

Fundamental Theorem of Arithmetic

- Fundamental theorem of arithmetic
  - Every number can be written uniquely as the product of primes

- $12 = 2^2 \times 3$ $30 = 2 \times 3 \times 5$ GCD(12, 30) = 2 x 3
- 2257 = 37*61 3337 = 47*71 GCD(2257, 3337) = 1

- $A = p_1^{\alpha 1} \times p_2^{\alpha 2} \ldots.$, $B = p_1^{\beta 1} \times p_2^{\beta 2} \ldots$
- $GCD(A,B) = p_1^{\min(\alpha 1, \beta 1)} \times p_2^{\min(\alpha 2, \beta 2)} \ldots$

GCD

- Proposed Algorithm for computing GCD(A,B)
  - Factor A and B and use the formula
  - Begs the question: how do you factor A?

- Proposed Algorithm to factor A
  - Try all prime numbers up to $\sqrt{A}$
  - What is wrong?

- A is exponential in lg A, the size of its representation!

Complexity of Factoring

- Does there exist a polynomial-time factoring algorithm?
- Major open question.

- Surprisingly, in a breakthrough 2002 result AKS (Agarwal, Kayal, Saxena) showed “Primes is in P”
- In other words, we know of a polynomial-time algorithm to tell that a number has non-trivial factors, without being able to determine what they are ?!?!

- Kayal and Saxena were undergraduates at the time!

GCD – Euclid's Algorithm

- Over 2 millenia ago the Greeks knew of a fast algorithm to compute GCD

- Entire algorithm in one line! Let $A \geq B$.
- GCD(A,B) = GCD(B, A mod B)

  - What is A mod B?
  - Remainder when A is divided by B, i.e. r where $A = qB + r$, $0 \leq r < B$

- GCD(30, 12) = (12, 6) = (6, 0) = 6 {Note: by convention (A,B) = GCD(A,B)}
- GCD(3337, 2257) = (2257, 1080) = (1080, 97) = (97, 13) = (13, 6) = (6, 1) = (1, 0) = 1

- Correctness & Complexity

Euclid's Algorithm - Correctness

Theorem: Euclid's Algorithm computes the GCD.

Proof:

Crux: if D divides A and B then D divides any linear combination Ax + By

Observe that A mod B = r = A – qB so if D divides A and B then D also divides B and (A mod B)

Also, A = qB + (A mod B) so if D divides B and A mod B then D also divides A and B

Hence, GCD(A, B) = GCD(B, A mod B).

QED

Euclid's Algorithm - Complexity

Theorem: Given n-bit numbers A, B GCD(A, B) is computable in $O(n^3)$ time

Proof:

Crux: after 2 steps the larger number is reduced by a factor 2

GCD(A, B) = GCD(B, A mod B) = GCD(A mod B, B mod (A mod B))

Case 1. $B > A/2$ then $A \bmod B < A/2$

Case 2. $B \leq A/2$ then $A \bmod B < A/2$

Either way the larger number is reduced by factor 2 after 2 steps so the algorithm terminates in $2 \lg A = O(n)$ steps. Taking the (naïve) complexity of mod (division) to be quadratic we get cubic complexity for Euclid.

QED

Summary

- GCD(A,B) is defined in terms of the prime factorization of A and B
- But, we don’t know how to factor efficiently, i.e. in polynomial-time
- With a bit of thought, we can bypass factoring entirely and compute GCD very quickly.
- Main Takeaway: Problems that appear hard at first blush can often be solved with a bit of cleverness.