CSL202: Discrete Mathematical Structures

Ragesh Jaiswal, CSE, IIT Delhi

Number Theory and Cryptography

Theorem

Let m be a positive integer. If $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$, then

$$a + c \equiv b + d \pmod{m} \quad and \quad ac \equiv bd \pmod{m}.$$

Theorem

Let m be a positive integer and let a and b be integers. Then

$$(a + b) \pmod{m} = ((a \pmod{m}) + (b \pmod{m})) \pmod{m}$$

and

$$ab \pmod{m} = ((a \pmod{m})(b \pmod{m})) \pmod{m}.$$

- Let $Z_m = \{0, 1, 2, ..., m-1\}$.
- We can define the following arithmetic operations on Z_m:
  - $+_m$: This is defined as $a +_m b = (a + b) \ (mod \ m)$.
  - $\cdot_m$: This is defined as $a \cdot_m b = (a \cdot b) \ (mod \ m)$.
- Show that $+_m$ and $\cdot_m$ satisfies the following properties:
  - Closure
  - Associativity
  - Commutativity
  - Identity
  - Additive inverse
  - Distributivity

Theorem

Let b be an integer greater than 1. Then if n is a positive integer, it can be expressed uniquely in the form

$$n = a_k b^k + a_{k-1} b^{k-1} + ... + a_1 b + a_0,$$

where k is a nonnegative integer, $a_0, a_1, ..., a_k$ are nonnegative integers less than b, and $a_k \neq 0$.

- What is the running time of each of the following operations:
  - Adding an m bit number with an n bit number.
  - Multiplying an m bit number with an n bit number.

Problem

Multiplying two n-bit numbers: Given two n-bit numbers, A and B, Design an algorithm to output $A \cdot B$.

Problem

Multiplying two n-bit numbers: Given two n-bit numbers, A and B, Design an algorithm to output $A \cdot B$.

- Solution 1: Use long multiplication.
- What is the running time of the algorithm that uses long multiplication?

Problem

Multiplying two n-bit numbers: Given two n-bit numbers, A and B, Design an algorithm to output $A \cdot B$.

- Solution 1: Use long multiplication.
- What is the running time of the algorithm that uses long multiplication? $O(n^2)$
- Is there a faster algorithm?

Problem

Multiplying two n-bit numbers: Given two n-bit numbers, A and B, Design an algorithm to output $A \cdot B$.

- Solution 1: Algorithm using long multiplication with running time $O(n^2)$.
- Solution 2: (Assume n is a power of 2)
  - Write $A = A_L \cdot 2^{n/2} + A_R$ and $B = B_L \cdot 2^{n/2} + B_R$.
  - So, $A \cdot B = (A_L \cdot B_L) \cdot 2^n + (A_L \cdot B_R + A_R \cdot B_L) \cdot 2^{n/2} + (A_R \cdot B_R)$
  - Main Idea: Compute $(A_L \cdot B_L)$, $(A_R \cdot B_R)$, and $(A_R \cdot B_L)$, and $(A_L \cdot B_R)$ and combine these values.

Problem

Multiplying two n-bit numbers: Given two n-bit numbers, A and B, Design an algorithm to output $A \cdot B$.

- Solution 1: Algorithm using long multiplication with running time $O(n^2)$.
- Solution 2: (Assume n is a power of 2)
  - Write $A = A_L \cdot 2^{n/2} + A_R$ and $B = B_L \cdot 2^{n/2} + B_R$.
  - So, $A \cdot B = (A_L \cdot B_L) \cdot 2^n + (A_L \cdot B_R + A_R \cdot B_L) \cdot 2^{n/2} + (A_R \cdot B_R)$
  - Main Idea: Compute $(A_L \cdot B_L)$, $(A_R \cdot B_R)$, and $(A_R \cdot B_L)$, and $(A_L \cdot B_R)$ and combine these values.

Algorithm

`DivideAndConquer`(A, B)
- If $(|A| = |B| = 1)$ return$(A \cdot B)$
- Split A into A_L and A_R
- Split B into B_L and B_R
- $P \leftarrow$ `DivideAndConquer`(A_L, B_L)
- $Q \leftarrow$ `DivideAndConquer`(A_R, B_R)
- $R \leftarrow$ `DivideAndConquer`(A_L, B_R)
- $S \leftarrow$ `DivideAndConquer`(A_R, B_L)
- return(`Combine`(P, Q, R, S))

- What is the recurrence relation for the running time of the above algorithm?

Problem

Multiplying two n-bit numbers: Given two n-bit numbers, A and B, Design an algorithm to output $A \cdot B$.

Algorithm

$\texttt{DivideAndConquer}(A, B)$
- If $(|A| = |B| = 1)$ return$(A \cdot B)$
- Split A into A_L and A_R
- Split B into B_L and B_R
- $P \leftarrow \texttt{DivideAndConquer}(A_L, B_L)$
- $Q \leftarrow \texttt{DivideAndConquer}(A_R, B_R)$
- $R \leftarrow \texttt{DivideAndConquer}(A_L, B_R)$
- $S \leftarrow \texttt{DivideAndConquer}(A_R, B_L)$
- return$(\texttt{Combine}(P, Q, R, S))$

- What is the recurrence relation for the running time of the above algorithm? $T(n) = 4 \cdot T(n/2) + O(n)$ for $n > 1$ and $T(1) = O(1)$.
- What is the solution to the above recurrence relation?

Problem

Multiplying two n-bit numbers: Given two n-bit numbers, A and B, Design an algorithm to output $A \cdot B$.

Algorithm

`DivideAndConquer`(A, B)
- If $(|A| = |B| = 1)$ return$(A \cdot B)$
- Split A into A_L and A_R
- Split B into B_L and B_R
- $P \leftarrow$ `DivideAndConquer`(A_L, B_L)
- $Q \leftarrow$ `DivideAndConquer`(A_R, B_R)
- $R \leftarrow$ `DivideAndConquer`(A_L, B_R)
- $S \leftarrow$ `DivideAndConquer`(A_R, B_L)
- return(`Combine`(P, Q, R, S))

- What is the recurrence relation for the running time of the above algorithm? $T(n) = 4 \cdot T(n/2) + O(n)$ for $n > 1$ and $T(1) = O(1)$.
- What is the solution to the above recurrence relation? $T(n) = O(n^2)$.

Problem

Multiplying two n-bit numbers: Given two n-bit numbers, A and B, Design an algorithm to output $A \cdot B$.

- Solution 1: Algorithm using long multiplication with running time $O(n^2)$.
- Solution 2: Naïve Divide and Conquer with running time $O(n^2)$.
- Solution 3:
  - Write $A = A_L \cdot 2^{n/2} + A_R$ and $B = B_L \cdot 2^{n/2} + B_R$.
  - So, $A \cdot B = (A_L \cdot B_L) \cdot 2^n + (A_L \cdot B_R + A_R \cdot B_L) \cdot 2^{n/2} + (A_R \cdot B_R)$
  - Main Idea: Compute $(A_L \cdot B_L)$, $(A_R \cdot B_R)$, and $(A_L + B_L) \cdot (A_R + B_R) - (A_L \cdot B_L) - (A_R \cdot B_R)$.

Problem

Multiplying two n-bit numbers: Given two n-bit numbers, A and B, Design an algorithm to output $A \cdot B$.

Algorithm

$\texttt{Karatsuba}(A, B)$

- If $(|A| = |B| = 1)$ return$(A \cdot B)$
- Split A into A_L and A_R
- Split B into B_L and B_R
- $P \leftarrow \texttt{Karatsuba}(A_L, B_L)$
- $Q \leftarrow \texttt{Karatsuba}(A_R, B_R)$
- $R \leftarrow \texttt{Karatsuba}(A_L + A_R, B_L + B_R)$
- return$(\texttt{Combine}(P, Q, R))$

- What is the recurrence relation for the running time of the above algorithm?

Problem

Multiplying two n-bit numbers: Given two n-bit numbers, A and B, Design an algorithm to output $A \cdot B$.

Algorithm

`Karatsuba`(A, B)

- If $(|A| = |B| = 1)$ return$(A \cdot B)$
- Split A into A_L and A_R
- Split B into B_L and B_R
- $P \leftarrow \texttt{Karatsuba}(A_L, B_L)$
- $Q \leftarrow \texttt{Karatsuba}(A_R, B_R)$
- $R \leftarrow \texttt{Karatsuba}(A_L + A_R, B_L + B_R)$
- return$(\texttt{Combine}(P, Q, R))$

- Recurrence relation: $T(n) \leq 3 \cdot T(n/2) + cn$; $T(1) \leq c$.
- What is the solution of this recurrence relation?

Problem

Multiplying two n-bit numbers: Given two n-bit numbers, A and B, Design an algorithm to output $A \cdot B$.

Algorithm

Karatsuba(A, B)
- If $(|A| = |B| = 1)$ return$(A \cdot B)$
- Split A into A_L and A_R
- Split B into B_L and B_R
- $P \leftarrow$ Karatsuba(A_L, B_L)
- $Q \leftarrow$ Karatsuba(A_R, B_R)
- $R \leftarrow$ Karatsuba$(A_L + A_R, B_L + B_R)$
- return(Combine(P, Q, R))

- Recurrence relation: $T(n) \leq 3 \cdot T(n/2) + cn$; $T(1) \leq c$.
- What is the solution of this recurrence relation?
  $T(n) \leq O(n^{\log_2 3})$

End