CSC006P1M: Design and Analysis of Algorithms
Lecture 10 *(Exponentiation, Euclid's Algorithm and Multiplicative Inverse)*

Sumit Kumar Pandey

September 12, 2022

Exponentiation

Exponentiation

Problem

Given two positive integers n and k, compute n^k.

First Attempt: $n^k = n^{k-1} \cdot n$.

ExpoV1(n,k)
Input: n and k (two positive integers)
Output: P
begin
    $P := n$;
    for $i := 1$ to $k - 1$ do
        $P := n * P$
end

$T(k) = \Theta(k)$.
$T(k) = \Theta(2^{\lg k})$.

Second Attempt:

- If k is even, $n^k = (n^{k/2})^2$.
- Else $n^k = (n^{\lfloor k/2 \rfloor})^2 * n$

Let $P(k) = n^k$.
Then $P(k) = P(\lfloor k/2 \rfloor)^2$ if k is even
else $P(k) = P(\lfloor k/2 \rfloor)^2 * n$.

Inductive Hypothesis

We know how to compute $P(\lfloor k/2 \rfloor)$.

Second Attempt:

ExpoV2(n,k)
Input: n and k (two positive integers)
Output: P

```
begin
    if k = 1 then P := n;
    else
        z := ExpoV2(n, ⌊k/2⌋);
        if k mod 2 = 0 then
            P := z * z;
        else
            P := n * z * z;
end
```

$T(k) = \Theta(\lg k)$.

Square and Multiply:

$k = \sum_{i=0}^{l-1} c_i 2^i$. Let $c = c_{l-1}, c_{l-2}, \cdots, c_1, c_0$

SQM(n,c)
Input: n, c
Output: P
begin
    $P := 1$
    for $i := l - 1$ to 0 do
        $P := P^2$;
        if $c_i = 1$ then
            $P := P \cdot n$;
end

$T(k) = \Theta(l)$.
$T(k) = \Theta(\lg k)$.

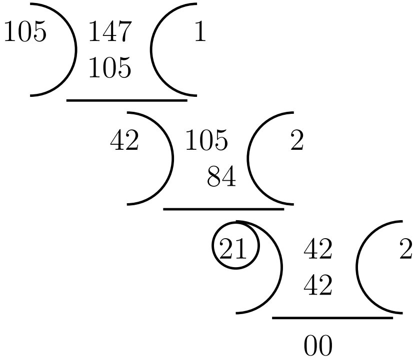
Square and Multiply:

$k = \sum_{i=0}^{l-1} c_i 2^i$. Let $c = c_{l-1}, c_{l-2}, \cdots, c_1, c_0$

- Number of Squares $= l$
- Number of Multiplications $= m$, where $1 \leq m \leq l$.
- Average number of Multiplications $\approx l/2$.

The Greatest Common Divisor

- $\gcd(105, 147)$



- $\gcd(105, 147) = 21$ (Why?)

The Greatest Common Divisor

$\text{gcd}(105, 147) =$

105) 147 (1
105
42) 105 (2
84
21) 42 (2
42
00

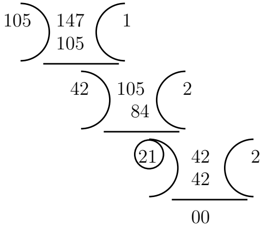
$\text{gcd}(42, 105) =$

42) 105 (2
84
21) 42 (2
42
00

$\text{gcd}(21, 42) =$

21) 42 (2
42
00

The Greatest Common Divisor



1. $\mathbf{147} = 1 \cdot \mathbf{105} + \mathbf{42}$
2. $\mathbf{105} = 2 \cdot \mathbf{42} + \mathbf{21}$
3. $\mathbf{42} = 2 \cdot \mathbf{21} + \mathbf{0}$

$$\begin{aligned} \gcd(105, 147) &= \gcd(105, 42) \\ &= \gcd(21, 42) \end{aligned}$$

The Greatest Common Divisor

Let $a = qb + r$ where $0 \leq r < |b|$.

$\gcd(a, b) = \gcd(b, r)$

GCD(a,b)
Input: a and b (two positive integers)
Output: G
begin
    if $b = 0$ then $G := a$;
    else
        if $a < b$ then swap(a,b);
        $G :=$ GCD(b, $a \mod b$);
end

$T(a, b) =?$.

Euclidean Algorithm: To find $gcd(a, b)$.

$$\begin{aligned}
a &= q_1 \cdot b + r_1, & 0 < r_1 < |b| \\
b &= q_2 \cdot r_1 + r_2, & 0 < r_2 < r_1 \\
r_1 &= q_3 \cdot r_2 + r_3, & 0 < r_3 < r_2 \\
&\vdots \\
r_{n-2} &= q_n \cdot r_{n-1} + r_n, & 0 < r_n < r_{n-1} \\
r_{n-1} &= q_{n+1} \cdot r_n + 0
\end{aligned}$$

$\text{gcd}(a, b) = r_n$.

The Greatest Common Divisor

$\gcd(12378, 3054) = 6$

$$\begin{aligned}
12378 &= 4 \cdot 3054 + 162 \\
3054 &= 18 \cdot 162 + 138 \\
162 &= 1 \cdot 138 + 24 \\
138 &= 5 \cdot 24 + 18 \\
24 &= 1 \cdot 18 + 6 \\
18 &= 3 \cdot \mathbf{6} + 0
\end{aligned}$$

$6 = 12378x + 3054y$. Find x and y.

$$\begin{aligned}
\mathbf{12378} &= 4\cdot\mathbf{3054}+\mathbf{162}\\
\mathbf{3054} &= 18\cdot\mathbf{162}+\mathbf{138}\\
\mathbf{162} &= 1\cdot\mathbf{138}+\mathbf{24}\\
\mathbf{138} &= 5\cdot\mathbf{24}+\mathbf{18}\\
\mathbf{24} &= 1\cdot\mathbf{18}+\mathbf{6}\\
\mathbf{18} &= 3\cdot\mathbf{6}+0
\end{aligned}$$

$$\begin{aligned}
\mathbf{6} &= \mathbf{24}-\mathbf{18}\\
&= \mathbf{24}-(\mathbf{138}-5\cdot\mathbf{24})\\
&= 6\cdot\mathbf{24}-\mathbf{138}\\
&= 6\cdot(\mathbf{162}-\mathbf{138})-\mathbf{138}\\
&= 6\cdot\mathbf{162}-7\cdot\mathbf{138}\\
&= 6\cdot\mathbf{162}-7\cdot(\mathbf{3054}-18\cdot\mathbf{162})\\
&= 132\cdot\mathbf{162}-7\cdot\mathbf{3054}\\
&= 132\cdot(\mathbf{12378}-4\cdot\mathbf{3054})-7\cdot\mathbf{3054}\\
&= 132\cdot\mathbf{12378}+(-535)\cdot\mathbf{3054}
\end{aligned}$$

$6 = 12378x + 3054y$; $x = 132$ and $y = -535$.

Euclidean Algorithm: To find $gcd(a, b)$.

GCD(a,b)
Input: a and b (two positive integers)
Output: G (the gcd of a and b)
begin
    if $a < b$ then swap(a,b);
    $r := 1$;
    while $r > 0$ do $\{r$ is the remainder$\}$
        $r := a \mod b$;
        $a := b$;
        $b := r$;
    $G := a$;
end

$T(a, b) = ?$.

Time Complexity:

Assume $a > b > 0$.

$$\begin{aligned} a &= q_1 \cdot b + r_1, \quad 0 < r_1 < b \\ b &= q_2 \cdot r_1 + r_2, \quad 0 < r_2 < r_1 \\ r_1 &= q_3 \cdot r_2 + r_3, \quad 0 < r_3 < r_2 \end{aligned}$$

$$\gcd(a, b) = \gcd(r_1, r_2)$$

What about the bit-sizes of a, b and r_1, r_2?

- bit-size$(r_1) \overset{?}{\leq}$ bit-size$(a)-1$?
- bit-size$(r_2) \overset{?}{\leq}$ bit-size$(b)-1$?

Time Complexity:

bit-size(r_1) $\overset{?}{\leq}$ bit-size(a)-1?

$a = q_1 \cdot b + r_1, 0 < r_1 < b$

Claim: $r_1 < a/2$.

- If $b \leq a/2$, then $r_1 < b \leq a/2$.
- If $b > a/2$, then $q_1 = 1$ and $r_1 = a - b < a - a/2 = a/2$.

Therefore,

bit-size(r_1) $\leq$ bit-size(a)-1.

- bit-size(r_1) $\leq$ bit-size(a)-1.
- bit-size(r_2) $\leq$ bit-size(b)-1.

In every two steps, input sizes decrease by at least 1. Therefore, after at most $2\lceil \lg a \rceil$ steps, the algorithm must stop. Hence, the running time $T(|a|, |b|) = O(\lg a)$ assuming $a \geq b > 0$.

The Multiplicative Inverse

The Multiplicative Inverse

$11^{-1} \mod 35$?

$11x \equiv 1 \mod 35$. Find x.

$x = 16$.

The Multiplicative Inverse

$$\begin{aligned} \mathbf{35} &= 3 \cdot \mathbf{11} + \mathbf{2} \\ \mathbf{11} &= 5 \cdot \mathbf{2} + \mathbf{1} \\ \mathbf{2} &= 2 \cdot \mathbf{1} + \mathbf{0} \end{aligned}$$

$$\begin{aligned} \mathbf{1} &= \mathbf{11} - 5 \cdot \mathbf{2} \\ &= \mathbf{11} - 5 \cdot (\mathbf{35} - 3 \cdot \mathbf{11}) \\ &= 16 \cdot \mathbf{11} - 5 \cdot \mathbf{35} \end{aligned}$$

$1 = 11x + 35y$; $x = 16$ and $y = -5$.

$11 \cdot 16 \equiv 1 \mod 35$; $11^{-1} \equiv 16 \mod 35$.

The Multiplicative Inverse

The linear congruence $ax \equiv 1 \mod b$ has a solution if and only if $\gcd(a, b) = 1$.

Example:

x exists for

- $7x \equiv 1 \mod 25.$
- $53x \equiv 1 \mod 101.$
- $34x \equiv 1 \mod 39$

x does not exist for

- $5x \equiv 1 \mod 25.$
- $52x \equiv 1 \mod 100.$
- $26x \equiv 1 \mod 39$

Euclidean Algorithm: $\gcd(r_0, r_1)$

$$
\begin{aligned}
r_0 &= q_1 \cdot r_1 + r_2, & 0 < r_2 < r_1 \\
r_1 &= q_2 \cdot r_2 + r_3, & 0 < r_3 < r_2 \\
&\vdots \\
r_{n-2} &= q_{n-1} \cdot r_{n-1} + r_n, & 0 < r_n < r_{n-1} \\
r_{n-1} &= q_n \cdot r_n
\end{aligned}
$$

Theorem

For $0 \leq j \leq n$, we have that $r_j = s_j r_0 + t_j r_1$, where the r_j's are defined as in the Euclidean Algorithm, and the s_j's and t_j's are defined in the recurrence below.

$$t_j = \begin{cases} 0 & \text{if } j = 0 \\ 1 & \text{if } j = 1 \\ t_{j-2} - q_{j-1} t_{j-1} & \text{if } j \geq 2 \end{cases}$$

and

$$s_j = \begin{cases} 1 & \text{if } j = 0 \\ 0 & \text{if } j = 1 \\ s_{j-2} - q_{j-1} s_{j-1} & \text{if } j \geq 2. \end{cases}$$

Theorem

For $0 \leq j \leq n$, we have that $r_j = s_j r_0 + t_j r_1$.

Proof by Mathematical Induction:

- Induction is on j. It is true for $j = 0$ and $j = 1$.
- We assume that the hypothesis is true for $j = k - 1$ and $k - 2$ where $k \geq 2$. So, we have $r_{k-2} = s_{k-2} r_0 + t_{k-2} r_1$ and $r_{k-1} = s_{k-1} r_0 + t_{k-1} r_1$.
- We now prove that it is true for $j = k$.

$$
\begin{aligned}
r_k &= r_{k-2} - q_{k-1} r_{k-1} \\
&= s_{k-2} r_0 + t_{k-2} r_1 - q_{k-1}(s_{k-1} r_0 + t_{k-1} r_1) \\
&= (s_{k-2} - q_{k-1} s_{k-1}) r_0 + (t_{k-2} - q_{k-1} t_{k-1}) r_1 \\
&= s_k r_0 + t_k r_1.
\end{aligned}
$$

MulInv(a, b)

Input: a and b
Output: t ($b^{-1} \mod a$ if exists, otherwise $\perp$)
begin
  $a_0 := a, b_0 := b, t_0 := 0, t := 1;$
  $q := \lfloor a_0/b_0 \rfloor, r := a_0 - qb_0;$
  while($r > 0$) do
    $temp := (t_0 - qt) \mod a;$
    $t_0 := t;$
    $t := temp;$
    $a_0 := b_0;$
    $b_0 := r;$
    $q := \lfloor a_0/b_0 \rfloor$
    $r := a_0 - qb_0$
  if $b_0 \neq 1$ then $t := \perp$
end

$$T(a, b) = O(\lg(a + b)).$$

Thank You