CSOR 4231, Section 3: Analysis of Algorithms I - Problem Set #6

Kangwei Ling - kl3076@columbia.edu

April 3, 2019

Collaborators

Zefeng Liu (zl2715), Kunyan Han (kh2931), Luoyao Hao (lh2913).

1 DPV 1.11

Note that $35 = 5 \times 7$, where $5, 7$ are primes. Let $4^{1536} - 9^{4824} = K$.

By Fermat's little theorem:

$$4^{1536} = (4^4)^{384} \equiv 1^{384} \equiv 1 \pmod 5$$

$$9^{4824} = (3^4)^{2412} \equiv 1^{2412} \equiv 1 \pmod 5$$

therefore:

$$4^{1536} - 9^{4824} \equiv 0 \pmod 5$$

it is divisible by 5. We can write $K = 5m$, where $m \in \mathbb{Z}$.

Also,

$$4^{1536} = (4^6)^{256} \equiv 1^{256} \equiv 1 \pmod 7$$

$$9^{4824} = (3^6)^{1608} \equiv 1^{1608} \equiv 1 \pmod 7$$

K is also divisible by 7. Thus $5m$ is divisible by 7. Since 5 and 7 are primes, m must be divisble by 7, so $K = 5m = 5 \times 7n, n \in \mathbb{Z}$.

Therefore $K = 4^{1536} - 9^{4824}$ is divisble by 35.

2 DPV 1.20

a $20 \mod 79$

by using the Extended Euclidean Algorithm, we have:

$$79 \cdot (-1) + 20 \cdot 4 = 1$$

so the inverse of 20 mod 79 is 4.

a 3 mod 62

by using the Extended Euclidean Algorithm, we have:

$$62 \cdot (-1) + 3 \cdot 21 = 1$$

so the inverse of 3 mod 62 is 21.

a 21 mod 91

21 has no inverse under modular 91 because they are not coprime (both divisible by 7).

a 5 mod 23

by using the Extended Euclidean Algorithm, we have:

$$23 \cdot 2 + 5 \cdot (-9) = 1$$

so the inverse of 5 mod 23 is -9.

3 DPV 1.21

All integers that is coprime to 11^3 have inverses modulo 11^3, since we can find the inverse via the Extended Euclidean Algorithm.

Since 11 is a prime number, it is obvious to see that if a number has no factor of 11, then it must be coprime to $11^3 = 1331$. Integers in range $[1, ..., 1331]$ that is divisible by 11 is $11 \cdot 1, 11 \cdot 2, 11 \cdot 3, ..., 11 \cdot 121$, therefore, in range $[1, ..., 1331]$, there are $1331 - 121 = 1210$ integers that has inverse under modulo 1331.

4 DPV 1.30

(a) We can use a Divide-and-Conquer method like mergesort to add n numbers each m bits long. There are $\log n$ levels, and at level i, there are 2^i numbers each $m + \log n - i$ bits long. At each level, the addition each two numbers run in parallel. The total depth of the circuit, thus is:

$$\log m + \log(m+1) + \log(m+2) + \cdots + \log(m + \log n)$$

If we assume n is not too large, such that $\log(m + \log n) = O(\log m)$, then it is $O(\log n \log m)$.

(b) We can express $x + y + z$ as $r + s$, where r is the result if we ignore the carry, and s is the carry result. To be more specific, the i th bit of r: $r_i = x_i \oplus y_i \oplus z_i$, and if at least two bits of x_i, y_i, z_i is 1, then $s_{i+1} = 1$, otherwise $s_{i+1} = 0$, and $s_0 = 0$.

By this method, the sum of $x + y + z$ is expressed as the sum of two number: $r + s$.

(c) Multiply two n-bit numbers is in the end the addition of n shifted numbers.

we can recursively group every 3 numbers and use the method in (b) to express them as the sum of just two binary number. We do this recursively until there are only two number left.

All circuits up to this point is in constant depth of level. And at the end we use a carry lookahead circuit to calculate $R + S$, which is of level $O(\log(2n)) = O(\log n)$ The recursive procedure of transformation from 3-sum to 2-sum has $O(\log n)$ levels, therefore it is of depth $O(\log n)$, and in the end the sum of the two number can be calculated with a circuit of depth $O(\log n)$. Summing up, we can multiplying two n-bit numbers with a circuit of depth $O(\log n)$.

5 DPV 1.37

(a) See the table below:

	mod 3	mod 5
0	0	0
1	1	1
2	2	2
3	0	3
4	1	4
5	2	0
6	0	1
7	1	2
8	2	3
9	0	4
10	1	0
11	2	1
12	0	2
13	1	3
14	2	4

Observations:

- the residues cycle.
- no two row have the same residues for both mod 3 and mod 5.

(b) WLOG, suppose for such pair (j,k), there exists $0 \leq i_1 < i_2 < pq$, s.t. $i_1 \equiv j \pmod{p}$ and $i_1 \equiv k \pmod{q}$, also $i_2 \equiv j \pmod{p}$ and $i_2 \equiv k \pmod{q}$.

Then let $d = i_1 - i_2$, we have $d \equiv 0 \pmod{p}$ and also $d \equiv 0 \pmod{q}$. We can write $d = mp, m \in \mathbb{Z}$. Since d is also divisible by q, but p and q are distinct primes, then m must be divisible by q, therefore $d = npq, n \in \mathbb{Z}$. So we derive that $d \equiv 0 \pmod{pq}$, i.e. $i_1 \equiv i_2 \pmod{pq}$.

However, i_1 and i_2 are both in the range $[0, pq)$, therefore it must be the case that $i_1 = i_2$, which contradicts our presumption that $0 \leq i_1 < i_2 < pq$.

(c) Let $w = j \cdot q \cdot (q^{-1} \mod p) + k \cdot p \cdot (p^{-1} \mod q)$. By definition, we know that $q \cdot (q^{-1} \mod p) = mp + 1$ and $p \cdot (p^{-1} \mod q) = nq + 1$, where $m, n \in \mathbb{Z}$.

Note that,

$$w \equiv j \cdot q \cdot (q^{-1} \mod p) \equiv jmp + j \equiv j \pmod{p}$$

Similarly,

$$w \equiv k \cdot p \cdot (p^{-1} \mod q) \equiv knq + k \equiv k \pmod{q}$$

Using the proof in (b), we know that $i = w \mod pq$.

(d) Parts (b) and (c) can be easily generalized to the case for n distinct primes $p_1, ..., p_n$.

For part (b): If $p_1 ..., p_n$ are n distinct primes, then for every n tuple $(k_1, ..., k_n)$ with $0 \leq k_i < p_i$, there is a unique integer $0 \leq h < \prod_i p_i$ such that $h \equiv k_i \mod p_i$ for $i = 1, 2, ..., n$.

The proof is also similar to (b), we suppose there exist h_1, h_2 that satisfy the requirement. The only difference is that at the end, we derive that $d = e \prod_i p_i, e \in \mathbb{Z}$, where we know that $h_1 \equiv h_2 \pmod{\prod_i p_i}$.

For part (c): the formula for general case should be,

$$i = \sum_i \{k_i \cdot (\prod_{j \neq i} p_j) \cdot ((\prod_{j \neq i} p_j)^{-1} \mod p_i)\} \mod \prod_i p_i$$

The proof is similar to (c), since for each i, we can write

$$k_i \cdot (\prod_{j \neq i} p_j) \cdot ((\prod_{j \neq i} p_j)^{-1} \mod p_i) = k_i \cdot (m_i p_i + 1)$$

thus reduce to the proof of generalized (b).

6 DPV 1.44

N_1, N_2, N_3 must be coprime to each other, otherwise if any two of them have common factor, we can calculate the gcd of the two and then factor any one of them into p_i, q_i, thereby obtained all secrets.

According to the problem definition, all three friends share the same e, we have,

$$\begin{aligned} M^3 &\equiv C_1 \pmod{N_1} \\ M^3 &\equiv C_2 \pmod{N_2} \\ M^3 &\equiv C_3 \pmod{N_3} \end{aligned}$$

where C_i is the ciphertext. With these, we can plug in the Chinese Remaider theorem in the previous problem. Here N_1, N_2, N_3 are not distinct primes, but rather distinct numbers that is coprime to each other. But the proof in DPV 1.37 still applies since N_1, N_2, N_3 have no common factor besides 1.

By the way we use the cryptosystem, the message M is split or padded to satisfy that $0 \leq M < N_i$, therefore $M^3 < N_1 N_2 N_3$.

Using the CRT in DPV1.34(d), we can get that,

$$\begin{aligned} M^3 =& \{C_1 \cdot (N_2 N_3) \cdot ((N_2 N_3)^{-1} \mod N_1) + C_2 \cdot (N_1 N_3) \cdot ((N_1 N_3)^{-1} \mod N_2) \\ &+ C_3 \cdot (N_1 N_2) \cdot ((N_1 N_2)^{-1} \mod N_3)\} \mod N_1 N_2 N_3 \end{aligned}$$

Then we can find the cube root of M^3 to recover M.

7 DPV 7.2

Let x_1, x_2, x_3, x_4 be the amounts of duckwheat to be transported from Kansas to NY, Kansas to CA, Mexico to NY, Mexico to CA respectively. Then the linear program can be written as:

$$\begin{aligned} \text{minimize} \quad & 2x_1 + 3x_2 + 4x_3 + x_4 \\ \text{subject to} \quad & x_1 + x_2 \leq 15 \\ & x_3 + x_4 \leq 8 \\ & x_1 + x_3 \geq 10 \\ & x_2 + x_4 \geq 13 \\ & x_1, x_2, x_3, x_4 \geq 0 \end{aligned}$$

For the inequalities, I assume that not all duckwheat must be transported from where it is produced, and more than required amount of duckwheat can be transported to where it is consumed.