01415 Computational Tools for Discrete Mathematics

Ran Wang   Day 04 Home Work

S111503
Ran Wang
14/09/2015

01415 Computational Discrete Math Fall 2015

Compulsory Homework Problems

1. Compute $x^2 R^{-1}$ mod m in two ways, and compare which is more efficient in terms of single-digit multiplication.

a. **Squaring algorithm** to compute x^2 followed by the **Montgomery reduction** algorithm to compute $x^2 R^{-1}$ mod m:

The single-digit multiplication of squaring is:

$$\frac{n+n^2}{2}$$

and of Montgomery reduction is:

$$n(n+1)$$

Noticeably, squaring of a n-digit number can at maximum be 2n digits or can at minimum be 2n-1 digits, which is exactly the same length as T in T R^{-1} mod m. Thus the total single-digit multiplication is:

$$\frac{n+n^2}{2}+n(n+1)=\frac{3}{2}(n^2+n)$$

b. Direct computation using **Montgomery multiplication** to compute compute xx R^{-1} mod m.

The single-digit multiplication in direct Montgomery multiplication is:

$$(2n^2+2n)$$

Therefore, in comparison with question a). , b). has much bigger coefficient at high order, so in terms of computationally efficiency, Montgomery multiplication is less efficient than the approach using squaring plus Montgomery Reduction.

2. How much faster is it to perform the **Montgomery multiplication** on a 64-bit computer than on a 32-bit computer?

Montgomery multiplication of two numbers is notated as Mont(x,y) = xyR^{-1} mod m (where R = b^n and n is the number of digits of x,y and m).

The register sizes effectively define the radix of of computation for iPhone 5 is $b_1=2^{32}$, and for iPhone 5s is $b_2=2^{64}$. In order to have the same radix b and digit length of the multipliers x,y to conduct Montgomery multiplication on both machines, for iPhone 5s, R = $b^n=(2^{64})^n=\quad(2^{32})^{2n}$. storing a n-digits number in radix b1 requires n digits, whereas it just requires n/2 digits for b2. Since the single digits multiplication for Montgomery multiplication is 2n(n+1), we have for each machine the single digits multiplication:

iPhone 5: Mont(x,y) = 2n(n+1)
iPhone 5s: Mont(x,y)= n(n/2+1)

the result is exactly the clock cycle on each machines, in order to know how much faster the one is than another, I make a quotient of them:

$$\frac{4n+1}{n+2}$$

When n->∞, we can conclude that iPhone 5s is roughly four times faster than iPhone 5.

3. Montgomery exponentiation
Firstly, we know the complexity of Montgomery multiplication is 2n(n+1), reviewing the algorithm I have Mont multiplication and single-digit multiplications:

Table 1 complexity of Montgomery Exponentiation in general

steps	**Mont**	**Single-digit Multiplication**
1	1	2l(l+1)
2	3t/2	3lt(l+1)
3	1	2l(l+1)

Step 1, Mont(g,R^2 mod p) with 1<=g<p, and R^2 mod p<p, we know that g and the remainder of R^2 mod p can be stored with l-digits radix b.

Step 2, there are 2 Mont multiplications, the first is imposed by the loop from t down to 0, the second is on the condition that when a_i=1. Since a=($a_{t-1,\dots,}a_0$) have t = 2048 bits, 1 and 0 is 50/50, we count it 1/2t Mont multiplication.

It is given that p is 2048 bits and b = 2^{64}, and in order to get R = b^l, we only need l =2048/64=32 to store the radix b for p. now we plug t and l into table 1 we have the 6492288 single digit multiplications in total.

Table 2 complexity of Montgomery Exponentiation

steps	**Mont**	**Single-digit Multiplication**
1	1	2122
2	3072	6488064
3	1	2122