CIS 3362 Quiz #4: Number Theory Solutions

Date: 10/25/2024

1) (6 pts) Determine the Prime Factorization of 1,518,735,400.

1,518,735,400 = 15187354 x 100 = 15187354 x 2^2 x 5^2, now divide out one more 2, and then continue with 3, 7, etc.

= 7593677 x 2^3 x 5^2

= 22139 x 2^3 x 5^2 x 7^3

= 131 x 2^3 x 5^2 x 7^3 x 13^2

= 2^3 x 5^2 x 7^3 x 13^2 x 131

Grading: 1 pt per prime factorized term (need both prime and exponent to get the point, extra bonus point for getting the problem fully correct)

2) (8 pts) Determine φ(1,518,735,400) **and give your answer in prime factorized form.**

$$\begin{aligned}\varphi(1{,}518{,}735{,}400) &= \varphi(2^3 \times 5^2 \times 7^3 \times 13^2 \times 131)\\ &= \varphi(2^3) \times \varphi(5^2) \times \varphi(7^3) \times \varphi(13^2) \times \varphi(131)\\ &= (2^3 - 2^2)(5^2 - 5)(7^3 - 7^2)(13^2 - 13)(131 - 1)\\ &= (2^2)(2^2 \times 5)(294)(13)(13 - 1)(130)\\ &= (2^4)(5)(2 \times 3 \times 7^2)(13)(2^2 \times 3)(2 \times 5 \times 13)\\ &\\ &= \underline{\mathbf{2^8 \times 3^2 \times 5^2 \times 7^2 \times 13^2}}\end{aligned}$$

Grading: 1 pt for splitting up phis
2 pts for plugging in all prime phi formulas
1 pt simplifying terms as non-primes
4 pts to gather everything for prime factorization

3) (6 pts) Determine the remainder when 16^{7623} is divided by 509. Note that 509 is prime. **For full credit use Fermat's Theorem.**

Note: $16^{508} \equiv 1 \pmod{509}$ via Fermat's Theorem.

$16^{7623} = 16^{(508)(15)+3} = (16^{508})^{15} \times 16^3 \equiv 1^{15} \times 4096 \equiv \underline{\mathbf{24 \pmod{509}}}$

Grading: 2 pts for exponent split
2 pts for properly plugging into Fermat's Theorem
2 pts for simplifying leftover part to proper remainder

4) (6 pts) Determine the remainder when 987^{249602} is divided by 27625. **For full credit use Euler's Theorem.**

$27625 = 25 \times 1105 = 125 \times 221 = 5^3 \times 13 \times 17$
$\phi(5^3 \times 13 \times 17) = (5^3 - 5^2)(13 - 1)(17 - 1) = 100 \times 12 \times 16 = 19200$
Thus, via Euler's Theorem, since gcd(987, 27625) = 1, $987^{19200} \equiv 1 \pmod{27625}$

$987^{249602} = 987^{(19200)(13)+2} = (987^{19200})^{13} \times 987^2 \equiv 1^{13} \times 974169 \equiv 7294 \pmod{27625}$

Grading: 3 pts phi of 27625
1 pts for exponent split
1 pts for properly plugging into Fermat's Theorem
1 pts for simplifying leftover part to proper remainder

5) (8 pts) Use the Fermat Factoring Method to factor 142127. Please fill out the table below. Note: More rows than necessary are provided.

x	x^2 - 142127	Perfect Square?
377	2	No
378	757	No
379	1514	No
380	2273	No
381	3034	No
382	3797	No
383	4652	No
384	5329	Yes, $5329 = 73^2$

So, factorization is $384^2 – 73^2 = (384 – 73)(384 + 73) = 311 \times 457$

142127 = **311** **x** **457**

Grading: 1 pt row 1, ½ pt for rows 2 – 7, 1 pt row 8, 2 pts writing as difference of squares factored, 1 pt for simplifying down to 311 and 457. (Round down for ½ pt.)

6) (5 pts) Prove that 2 is a generator mod 11 the slow way, by listing the values of 2 raised to each power from 1 to 10 mod 11 below. (Note: Only the answers will be graded, so for once no work has to be shown.)

pow	1	2	3	4	5	6	7	8	9	10
2^{pow} mod 11	2	4	8	5	10	9	7	3	6	1

Grading: ½ pt per slot, round down

7) (10 pts) On the next page you'll complete a function that takes in a positive integer, ***n*** ($1 < n \leq 10{,}000$) and a pointer to an integer, and returns an array storing all of the integers in between 1 and ***n-1*** that are relatively prime with ***n***. (The function will also store the size of the array, $\phi(n)$ in the integer pointed to by the second parameter.) Complete the function so it works in this particular manner.

1. Create an integer array, **`relprime`**, of size ***n***, storing 1 in each index.. After completing step 2, relprime[i] = 1 if gcd(n, i) = 1 and relprime[i] = 0 if gcd(n, i) > 1. **This is done for you.**

2. Loop through all the integers from i = 2 to n-1, and for each integer i that is a divisor of n, find each of the **non-negative** multiples, x, of those values of i, and set relprime[x] = 0.

3. After step 2, count how many values in relprime are set to 1. As you are counting those values, store the indexes where 1 is stored in a new array, **`res`**, in numerical order. (Thus, res will store the numbers relatively prime to n.) Also, update the value of the integer pointed to by szPtr, the

second paramter. At the end resize the array res and return it. **(Note: A good portion of this step is done for you.)**

```
int* getRelPrime(int n, int* szPtr) {

    int* relprime = malloc(sizeof(int)*n);
    for (int i=0; i<n; i++) relprime[i] = 1;

    // Grading: 1 pt
    for (int i=2; i<n; i++)

        // Grading: 2 pts
        if (n%i == 0)

            // Grading: 3 pts, only 1 pt if they jump by 1 and
            // use an if.
            for (int j=0; j<n; j+=i)

                // Grading: 2 pts.
                relprime[j] = 0;


    int* res = calloc(n,sizeof(int));
    *szPtr = 0;
    for (int i=0; i<n; i++)

        if ( relprime[i] ) // Grading: 1 pt all or nothing

            res[(*szPtr)++] = i; // Grading: 1 pt all or nothing

    res = realloc(res, sizeof(int)*(*szPtr));
    return res;
}
```

8) (1 pts) What animal is in the logo of the local eatery, Pig Floyd's?

Pig (Grading: Give to All)