Competitive Programming

Factoring and GCD

Factoring, Primes and GCD

- Common task or sub-task for problems
- Chance to work with numbers larger than 32 bits

Factoring … the dumb way

```
void factor( long val, vector< long > factors ) {
  // Try all possible factors
  for ( long f = 1; f <= val; f++ )
    // See if f divides val
    if ( val % f == 0 )
      factors.add( f );
}
```

This will be too slow, if val is 10^9 or greater.

This divide operation can be a little expensive.

- Multiply
  - 10^9 int multiply operations : about the same cost as addition
  - 10^9 long multiplies : about the same cost as addition
- Divide
  - 10^9 int divide or mod operations : 1.7 seconds
  - 10^9 long divide or mod operations : 7 seconds

(most) Factors come in pairs

```
void factor( long val, vector< long > factors ) {
  for ( long f = 1; f * f <= val; f++ )
    if ( val % f == 0 ) {
      factors.add( f );
      factors.add( val / f );
    }
}
```

Get the small factor.

Get the larger factor.

Getting factors in pairs

```
void factor( long val, vector< long > factors ) {
  for ( long f = 1; f * f <= val; f++ )
    if ( val % f == 0 ) {
      factors.add( f );
      factors.add( val / f );
    }
}
```

Two copies of the same factor if val is a perfect square

(easy to fix)

Prime Factorization

```
void primeFactor( long val, vector< long > factors ) {
  long f = 2;
  while ( f * f <= val ) {
    while ( val % f == 0 ) {
      factors.add( f );
      val /= f;
    }

    f += 1;
  }

  if ( val != 1 )
    factors.add( val );
}
```

Save the factor and divide it out.

Whatever’s left in val must be prime.

Greatest Common Divisor (recursive)

```
long gcd( long a, long b ) {
  if ( a % b == 0 )
    return b;
  else
    return gcd( b, a % b );
}
```

Greatest Common Divisor (iterative)

```
long gcd( long a, long b ) {
  while ( b != 0 ) {
    long t = b;
    b = a % b;
    a = t;
  }
  return a;
}
```

Least Common Multiple

```
lcm = a * b / gcd( a, b );
```

This could overflow (even if the LCM doesn't)

Least Common Multiple

```
lcm = a * b / gcd( a, b );
```

```
lcm = a / gcd( a, b ) * b;
```

This could be safer.