Some major arithmetic ideas and results from MA/CSSE 473\

1. Adding two k-bit integers: $\Theta(k)$
2. Multiplication of two k-bit integers:
    a. Standard approach: $\Theta(k^2)$
    b. Simple divide and conquer (a+bi)(c+di) $\Theta(k^2)$
    c. Gauss-based divide-and conquer (a+b)(c+d) = (ac + bd) +[(a +b)(c+d) – ac – bd]I $\Theta(k^{\log[2]3})$
3. Division of two k-bit integers: $\Theta(k^2)$
4. Modular Arithmetic basics: $a \equiv b \pmod{N}$ if and only if N divides (a-b). I.e. $\exists k\ ((b-a) = kN$.
    a. Substitution rule
        i. If $x \equiv x' \pmod{N}$ and $y \equiv y' \pmod{N}$,
           then $x + y \equiv x' + y' \pmod{N}$, and $xy \equiv x'y' \pmod{N}$
    b. Associativity
        i. $x + (y + z) \equiv (x + y) + z \pmod{N}$
    c. Commutativity
        i. $xy \equiv yx \pmod{N}$
    d. Distributivity
        i. $x(y+z) \equiv xy + yz \pmod{N}$
    e. Modular addition run time $\Theta(k)$
    f. Modular multiplication run time $\Theta(k^2)$
    g. Integer division algorithm (gives quotient and remainder) $\Theta(k^2)$
    h. ModExp calculates $x^y \pmod{N}$ $\Theta(k^3)$
    i. Euclid algorithm: gcd(a, b) = gcd(b, a%b)
    j. Extended Euclid finds gcd d and x, y such that d = a x + b y.
    k. Modular inverse. X such that $ax \equiv 1 \pmod{N}$. Exisist iff gcd(a, N) = 1.
        i. Once we find x, y such that 1 = a x + b N, then $a^{-1} \equiv x \pmod{N}$
    l. Modular division $a/b \equiv a\, b^{-1} \pmod{N}$
5. Fermat's little theorem: If p is prime and p does not divide a, $a^{p-1} \equiv 1 \pmod{p}$
6. This test can show a number composite but cannot show it to be prime.
7. If a is relatively prime to N and if a fails the Fermat test (a^{p-1} is not congruent to 1 mod N), then at least half of 1, 2, …, N-1 fail the test.
8. A Carmichael number is a composite integer N for which each of 1, 2, …N-1 passes the Fermat test.
9. Miller-Rabin test. Write N-1 as $2^t u$ and Examine powers of a: u, u^2, u^4, … u^t looking for a nontrivial square root of 1. If we find one (or if the final power is not 1, so a fails the Fermat test) , N is composite.
10. RSA encryption: Let p and q be primes, N=pq. Let we be any number with gcd(e, N) = 1.
    a. The public encryption key is the pair (N, e)
    b. Encryption: $m' = m^e \pmod{N}$
    c. Decryption key d is the inverse of e (mod (p-1)(q-1)).
    d. Decryption: $(m')^d \equiv m \pmod{N}$. i.e. $(M^e)^d \equiv m \pmod{N}$.
    e. Principle. It's hard to guess d if you don't know p and q. Factoring is hard.