Lecture 36, Nov. 15

36.1 Example. Solve

$$\begin{aligned} 5x &= 9 \mod 14 \\ 7x &= 4 \mod 15 \end{aligned}$$

Solution. Euclidean Algorithum

$$\begin{aligned} 14 &= 2 \times 5 + 4 \\ 5 &= 1 \times 4 + 1 \\ &= 1 \times (14 - 2 \times 5) + 1 \\ 3 * 5 &= 1 \times 14 + 1 \end{aligned}$$

Let $x = 14k + 13$, then

$$\begin{aligned} 7(14k + 13) &= 4 \mod 15 \\ 8k &= 3 \mod 15 \end{aligned}$$

By inspection, $k = 2 \times 8k = 3 \mod 15$, then $k = 15t + 6$

$$\begin{aligned} x &= 14(15t + 6) + 13 \\ &= 210t + 97 \end{aligned}$$

Thus $x = 97 \mod 210$ is the solution

Cryptography

Primality Test Given an integer p, determine if p is prime.

36.2 Example (Trial Division). $\forall 2 \leq d \leq \sqrt{p}$, if $\exists d \mid p$, then p is composite. Otherwise p is prime.

36.3 Definition (Algorithm Efficiency). We call $f(n) \in O(g(n))$ if $\exists M, N \forall n > N \ f(n) \leq Mg(n)$.

36.4 Definition (Efficient). An algorithm is efficient if its worst-case running time on n-bit input is $O(n^k)$ for some k. (Note: The original way of finding if p is prime is growing exponentially, but we want polynomial growth to be "efficient")

36.5 Example. Input: a, b n-bit integer Output:

$$\begin{aligned} a + b \\ a = (a_n + 1.....a_0)_2 \\ b = (b_n + 1.....b_0)_2 \end{aligned}$$

Each bit take at most 2 ops. In total at most 2n ops which takes O(n) time. Which means that the multiplication of the prime number of take $O(n^2)$ time.

36.6 Algorithm (Repeated Square Algorithm).

$$a^k = \prod a^{2^i} \mod n$$

36.7 Example.

$$
\begin{aligned}
3^{13} \quad & \mod 19 \\
13 &= 2^3 + 2^2 + 1 \\
3 &= 3 \mod 19 \\
3^2 &= 9 \mod 19 \\
3^4 &= 81 \mod 19 \\
&= 5 \mod 19 \\
3^8 &= 5^2 = 25 = 6 \mod 19 \\
3^{13} &= 3^8 3^4 3^1 = 14 \mod 19
\end{aligned}
$$