Testing Primality

THESE NOTES ARE CRYPTIC AND TERSE. FURTHER DETAILS ARE FOUND IN THE CHAPTER HANDED OUT IN CLASS.

We can test if two numbers are coprime efficiently using Euclid's algorithm. We can test primality using brute force or sieve approaches but not in polynomial time. But we can test if a number has a small factor.

We consider the multiplicative group Z_n^* of the numbers relatively prime to n. Let the size of this group be $\phi(n)$. The *order* of an element a in this group is the minimum k such that a^k is 1. By partitioning the group we see that the order of a must divide the order (i.e. size) of the group. This gives us Euler's Theorem and Fermat's Theorem (see handout).

How does this compare with composite numbers?? For any n let F_n be the set of residues satisfying $x^{n-1} = 1 \ mod \ n$. One can show that for all composite numbers less then 561, $|F_n| \leq \frac{1}{2}|Z_n|$. Unfortunately can show that there are infinitely many bad composite numbers for which this property does not hold. If we could somehow deal with these we would have a fast algorithm because of fast exponentiation.

We will show (third step below) that Z_n^* is cyclic for prime n. Thus, there are $\frac{n-1}{2}$ residues which have no square roots, by Fermats theorem, while everybody else has at least two and hence exactly two square roots. It follows that the square roots of 1 are 1 and -1 and hence for non-squares x, $x^{\frac{p-1}{2}} = -1$. by Fermat's theorem.

We next discussed the Legendre symbol and how to compute it.

We then defined the Jacobi symbol. We will show how to compute it.

We will show(step 2 below): Theorem: If n is odd and not prime then the set Jn of residues with Jacobi symbol equal to their $(\frac{n-1}{2})$th power is at most 1/2 of Z_n..

Our algorithm consists of choosing k elements uniformily at random from Z_n and checking if their Jacobi symbol is equal to their $\frac{n-1}{2}$th power.

First step:- how to compute it See chapter.

Second step: Just need to show that it is a proper subgroup. See Chapter

Third step: Lemma 14.12 and Theorem 14.13 not too bad. May leave the end til the next class.

Theorem 4.29 and algorithm.