# Factoring from Order Finding

Given that we efficiently find the order of elements in $\mathbb{Z}_d^*$, we can also efficiently factor large integers!

Consider factoring an arbitrary integer $x$ of $n$ bits

- We can efficiently check if $x$ is prime. $O(n^{12})$ Agrawal, Kayal, Saxena
- We can efficiently factor $x$ if $x = p^k$ for prime $p$, so just running this algorithm and checking the output is enough to deal with this case.
  (compute the first $n = \log_2 x$ roots of $x$, check if any are integers and prime with AKS algorithm above)
- Evenness is also easy to check: divide by 2. We can say this about any fixed set of primes, of course, but checking each doesn't scale well as $x$ gets larger, so we just choose to check 2.
- Clearly, we really only need an algorithm to spit out a single number $s$ dividing $x$, since then we've got $s$ and $x/s$, each of which we subject to primality testing, prime power factoring, and further factoring if necessary.
- We can also compute the greatest common divisor of two $n$ bit numbers in $O(n^2)$ steps using the Euclidean algorithm

Here's how to find $s$ and $t = x/s$ using order finding:

1. Randomly choose $a \in \{2, \ldots, x-1\}$
2. Compute $d = \text{GCD}(a, x)$
3. If $d \geq 2$ then return $s = d$, $t = x/d$ (this won't happen often...)
4. Otherwise: (now we know $a \in \mathbb{Z}_x^*$)
   a. Find $r$ such that $a^r \bmod x = 1$
   b. If $r$ is even, then
      i. compute $y = a^{r/2} - 1 \bmod x$
      ii. compute $d = \text{GCD}(y, x)$
      iii. if $d \geq 2$, return $s = d$, $t = x/d$

Repeat as needed.

The magic is all in section 4b, clearly. (Why) does it work? Since $a^r = 1 \bmod x$, $x$ divides $a^r - 1$, and therefore it divides $(a^{r/2}+1)(a^{r/2}-1)$. Thus the factors of $x$ are somehow distributed between $a^{r/2}+1$ and $a^{r/2}-1$. They cannot be completely contained in the factors of $a^{r/2}-1$ alone, or else $x$ would divide $a^{r/2}-1$, and the order of $a$ would be $r/2$. If $x$ also does not divide $a^{r/2}+1$, then $a^{r/2}-1$ and $x$ must have some factor(s) in common, so we can compute $\text{GCD}(a^{r/2}-1, x)$ and find the largest one. Repeating this process on the results will eventually give the prime factorization of $x$.