Lecture 16

Remark 3. In the case of $f \in D_1$, $\Pr[\mathrm{rk}(A) = n-1] \geq 1 - 2^{n-1-K}$ by a similar argument because, in this case, the rows of A are chosen uniformly from the subspace orthogonal to s, which has dimension $n-1$. Therefore, if $K = n + 100$, A is almost certainly of rank $n-1$. In this case, since $As = 0$, the kernel of A must be $\{0, s\}$, where s is the period of f. Then, s can be recovered from A by computing the kernel of A and outputting the non-zero element.

Comment: Do example with the rank-2 matrix above, directly. Note that can be more systematically done by Gaussian elimination and converting A to reduced row echelon form $\tilde{A}$. Then the kernel can be read off by expressing the pivot variables in terms of the non-pivot variables. For a reminder of how this works, see this Brilliant wiki page (which does linear algebra over $\mathbb{R}$ but the same method works over $\mathbb{F}_2$).

Factoring and Shor's algorithm.

Definition 15 (Prime and composite numbers)**.** A prime number is a positive integer $p \geq 2$ that has no positive divisors other than 1 and p. A positive integer that is not prime is called composite.

Theorem 1 (Fundamental theorem of arithmetic)**.** *Every positive integer can be factored into a product of primes. Moreover, the factorization is unique (up to reordering).*

Remark 4. Is the theorem obvious? No for the first part, because it's about primes dividing numbers but the definition of prime is about numbers dividing primes. No for the second part, because it's not obvious that I can decide

$$31 \times 7 \neq 17 \times 11, \tag{86}$$

by observing $\{31, 7\}$ and $\{17, 11\}$ are different multisets of primes.

Prime factorization problem: given a composite positive integer N, find its prime factors.

Definition 16 (Modulo N operation.)**.** Let N be a positive integer. Given integers $x, y \in \mathbb{Z}$,

1. write $x \mod N$ to mean the unique element x' in $\{0, 1, \ldots, N-1\}$ such that N divides $x - x'$.
2. write $x = y \mod N$ if $x \mod N = y \mod N$, and $x \neq y \mod N$ otherwise.

Example 2. *Take $N = 10$, $x = -2$, then $x \mod 10 = 8$. Take $y = 24$, then $y \mod 10 = 4$ so $x \neq y \mod 10$.*

Definition 17 (GCD and coprimality)**.** Given integers a, b not both zero, the greatest common divisor of a and b is the largest positive integer d such that $d \mid a$ and $d \mid b$, and is denoted $\gcd(a, b)$. We say that a, b are coprime if $\gcd(a, b) = 1$.

Fact 6. Let a, N be positive integers. If a and N are coprime, then there exists a positive integer r such that $a^r = 1 \mod N$.

Comment: proof involves pigeon hole principle and Bezout's theorem to get the inverse of a modulo N.

Definition 18 (Order of a mod N)**.** Let a, N be positive integers that are coprime, the order of a modulo N, denoted $\mathrm{ord}_N(a)$, is the *smallest* positive integer r such that $a^r = 1 \mod N$.

Comment: $\mathrm{ord}_N(a)$ agrees with the order of a in $\mathbb{Z}_N^*$ in the group-theoretic sense ($\mathbb{Z}_N^*$ is the multiplicative group of integers modulo N").

N is composite and has two distinct odd prime factors. For the general case, see my lecture notes here or the other resources listed on the course website. Recall that Shor's algorithm runs in time $O(\mathrm{poly}(\log(N)))$ whereas the best known classical algorithm takes time $2^{O(\log^{1/3}(N))}$.

Given such N, the algorithm works as follows. The only quantum computing part is highlighted in red. Everything else is done on a classical computer.

1. Choose a uniformly at random from the set $\{1, \ldots, N-1\}$. Compute $d \coloneqq \gcd(a, N)$. If $d > 1$, then output d. Else d must be 1 (so a, N are coprime) and we continue.
2. Compute $r \coloneqq \mathrm{ord}_N(a)$ on a quantum computer. If r is odd, output: "FAIL", else continue.
3. Compute $d' \coloneqq \gcd(a^{r/2} - 1 \mod N, N)$. If $d' > 1$, then output d', otherwise output "FAIL".